TECHNOLOGY DEPT.

COMMUNICATIONS

Of The Association For

COMPUTING MACHINERY

Volume 1 . Number 5 May 1958

PUBLIC LIBRARY
JUN 13 1958

Official Notices 1

Techniques 3

Unusual Applications 14

News and Notices 21

Published by the Association for Computing Machinery

COMMUNICATIONS OF THE

ASSOCIATION FOR COMPUTING MACHINERY

Editor-in-Chief
ALAN J. PERLIS
Computation Center
Carnegie Institute of Technology
Pittsburgh 13, Pennsylvania

News and Notices
WALTER F. BAUER
Guided Missile Research Division
The Ramo-Wooldridge Corp.
P. O. Box 45564 Airport Station
Los Angeles 45, California

Standards
GRACE M. HOPPER
Remington Rand Univac Division
Sperry Rand Corp.
19th and Allegheny Avenue
Philadelphia 40, Pennsylvania

Managing Editor
SAUL ROSEN
Philco Corporation
Government and Industrial Div.
4700 Wissahickon Avenue
Philadelphia 44, Pennsylvania

Techniques
ROBERT W. BEMER
World Headquarters
International Business Machines Corp.
590 Madison Avenue
New York, New York

Unusual Applications
ARTHUR WOUK
Electronic Systems Division
Sylvania Electric Products, Inc.
100 First Avenue
Waltham, Massachusetts

COMMUNICATIONS OF THE ASSOCIATION FOR COMPUTING MACHINERY is published monthly for the Association for Computing Machinery, 5800 N. Marvine St., Philadelphia 41, Pa. Second-Class mail privileges authorized at Philadelphia, Pennsylvania.

Subscription price per year $2.00 for members, $5.00 for non-members. Single copies of Communications $1.00. Subscriptions, notices of address changes and other communications on business matters should be addressed to the Association for Computing Machinery, 2 East 63rd Street, New York 21, N. Y. Material intended for publication and communications on editorial matters should be sent in duplicate to the Editor-in-Chief, or to the Editor of the appropriate department.

By arrangement with the office of Naval Research, U. S. Navy Dept., the Digital Computer Newsletter issued quarterly by that organization is reprinted in its entirety as a supplement to the Communications.

OFFICIAL NOTICES

Back Numbers of ACM Journal and Proceedings

Some of the early issues of the ACM Journal are out of print; of others a small supply is still on hand. The Association is negotiating with a commercial publisher for reprinting the early volumes. When this is done, the existing stock will have to be withdrawn, and single issues may not be available at all, or at least not at today's prices. If some issues are missing in your collection, this may be your last chance to acquire them.

Available:

ACM Journal, vol. 1, no. 1 $2.50
vol. 2, nos. 1–4 2.50 each
vol. 3, nos. 2–4 2.50 each
Proceedings, ACM. Sept. 1952 (Toronto) $3.00

Out of print:

ACM Journal, vol. 1, nos. 2–4
vol. 3, no. 1
Proceedings, ACM, May 1952 (Pittsburgh)

PROGRESS REPORT ON ORGANIZING FOR THE FIRST INTERNATIONAL CONFERENCE ON INFORMATION PROCESSING

It is with considerable pleasure that we are able to report to you that the proposal initiated by the National Joint Computer Committee of the United States to organize an International Conference on Information Processing has been included in the draft program and budget of UNESCO for 1959–60, which will be presented to the UNESCO Executive Committee in April 1958.

On December 18, 1957, the UNESCO Department of Natural Sciences, at the initiative of Professor Pierre Auger, convened a conference of consultants and observers to advise the Director-General of UNESCO on plans for convening the First International Conference on Information Processing.

For your information and planning in connection with participation and attendance at this conference, there are summarized below the more important recommendations that were formulated:

a. *Title:* First International Conference on Information Processing (French title: Premiere Conference Internationale aur le Traitement des Donnees Scientifiques)

b. *Time:* June 15–20, 1959

c. *Place:* Paris or Rome (decision to be made by mid-1958)

d. *Sponsors:* UNESCO and those national technical societies that include computer interests.

e. *Technical Societies:* To initiate the necessary preparations, the following national technical societies have agreed to cooperate:

United States.....National Joint Computer Committee
Germany.....Deutscher Ausschuss Fuer Rechnen Anlagen (DARA)
Japan.....Computing Division of the Science Council of Japan
France.....Association Francaise de Calcul
Italy.....Consiglio Nazionale delle Ricerche
USSR.....Academy of Sciences
United Kingdom....The British Group for Computation and Automatic Control of the British Conference on Automation and Computation (of which the British Computer Society is a member)

f. *Conference Subjects:*

1. Methods of digital computing.
2. Logical design and common symbolic language for machines.
3. Automatic translation of languages.
4. Collection, storage and retrieval of information.
5. Pattern recognition and machine learning.

g. *Exhibits:* A major exhibition of equipments will be held at the Conference. The exhibition will open a few days prior to the initial plenary sessions of the conference and will close a few days after the last plenary session. For more information, contact either your national technical society or UNESCO's Department of Natural Sciences after May 1958.

h. *Participants:* Participation will be on an individual rather than national basis, in a manner to be promoted by the national societies.

The NJCC takes this opportunity to express our appreciation to you for your assistance through your various technical societies which has led to adoption of this project by UNESCO. We believe that a most worthwhile step has been achieved in opening an international communication link for people working in the computer field. We look forward to seeing you and the other members of the computer society at the meeting in June 1959.

Thank you again for your support to date.

Isaac L. Auerbach, Chairman

for the
NJCC Subcommittee to Organize an
International Conference on Information Processing

Isaac L. Auerbach . . . . . Auerbach Electronics Corporation
109 North Essex Avenue
Narberth, Pa.

Samuel N. Alexander . . . . . National Bureau of Standards
Conn. Ave. and Van Ness St.
Washington 25, D. C.

Alston S. Householder . . . . . Oak Ridge National Laboratory
Oak Ridge, Tennessee

Please address correspondence to

Miss Margaret R. Fox, Secretary
National Joint Computer Committee
Box 4999, Washington 8, D. C.

USCOMM-NBS-DC

TECHNIQUES DEPARTMENT

NOTE ON EMPIRICAL BOUNDS FOR GENERATING BESSEL FUNCTIONS

JAMES B. RANDELS and ROY F. REEVES, Ohio State University, Columbus, Ohio

Introduction

It is well known that the use of the formula

$$(1) \qquad J_{n+1}(x) = \frac{2n}{x} J_n(x) - J_{n-1}(x)$$

to generate Bessel's function of the first kind for a fixed x from the starting values $J_0(x)$ and $J_1(x)$ works quite satisfactorily so long as n is small relative to x. However, if n is larger than x, the computation becomes unstable and the error grows rapidly out of hand. A method of circumventing this difficulty has been known in limited circles for some time [1, 2], but its use has not been widespread. The purpose of this note is to call attention to the method, its simplicity and efficiency, and to present some experimentally determined bounds on n relative to x for which the method is applicable.

The Method

The method consists of choosing an $\hat{n}$ sufficiently large relative to x (see next section), assuming $J^*_{\hat{n}+1}(x) = 0$ and $J^*_{\hat{n}}(x) << 1$ but $\neq 0$, and then using the recursive formula

$$(2) \qquad J^*_{n-1}(x) = \frac{2n}{x} J^*_n(x) - J^*_{n+1}(x)$$

to generate the set of values $J^*_n(x)$, $n = \hat{n}, \hat{n}-1, \cdots, 2, 1$. The $J^*_n(x)$ differ from the true $J_n(x)$ by a multiplicative constant, i.e., $J_n(x) = KJ^*_n(x)$, $n = 0, 1, \cdots, \hat{n}$.

This constant may be determined from the series

$$(3) \qquad K = J^*_0(x) + 2 \sum_{n=1}^{\hat{n}} J^*_{2n}(x).$$

Once the numbers $J_n(x)$, $\hat{n} = 0, 1, \cdots, \hat{n}$ have been computed, the problem of computing $Y_n(x)$, $n \geq 0$ is quite simple, since $Y_0(x)$ and $Y_1(x)$ can be found using the formulas

$$(4) \qquad Y_0(x) = \frac{2}{\pi} \left[J_0(x) \left(\gamma + \ln \frac{x}{2} \right) - 2 \sum_{n=1}^{\hat{n}} (-1)^n J_{2n}(x) \right], \qquad \gamma = .57721566 \text{ (Euler's constant)}$$

and

$$(5) \qquad Y_1(x) = \frac{J_1(x)\, Y_0(x) - \dfrac{2}{\pi x}}{J_0(x)},$$

The recursive relation

$$(6) \qquad Y_{n+1}(x) = \frac{2n}{x} Y_n(x) - Y_{n-1}(x)$$

provides a stable means of computing the high ordered values of $Y_n(x)$.

Since the range of the numbers used in the above method is very large, floating point operations are necessary to carry out the actual calculations.

Empirical Bounds For Choosing $\hat{n}$ In The Generation of $J_n(x)$

Before one may apply this method, it is necessary to determine an $\hat{n}$ (or a range of values of $\hat{n}$) which will guarantee correct results. One might immediately suspect that if $\hat{n}$ is chosen too small relative

[1] British Association Math. Tables X, *Bessel Functions, Part II*, University Press, Cambridge, 1952, pg. xvii.
[2] Private communication with Milton Abramowitz, National Bureau of Standards.

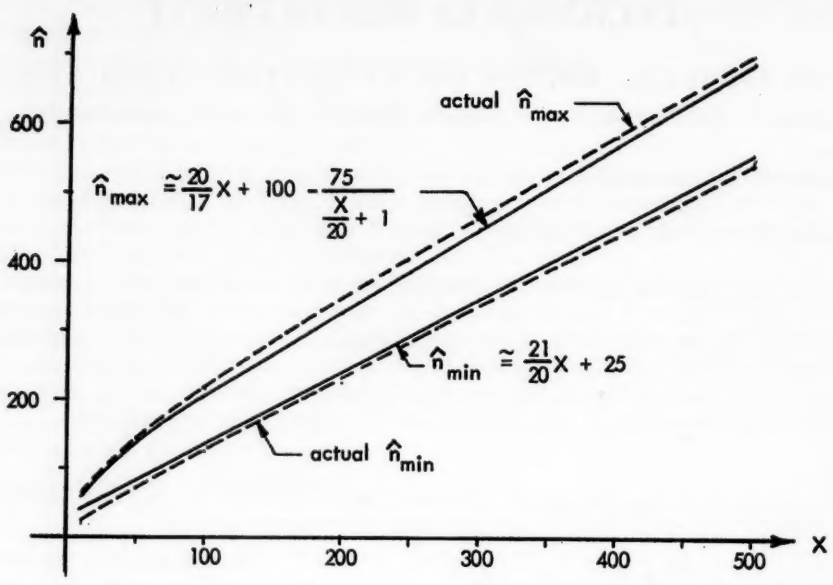


Figure 1. Range of Acceptable Values of $\hat{n}$ relative to X.
$10 \leq X < 500$

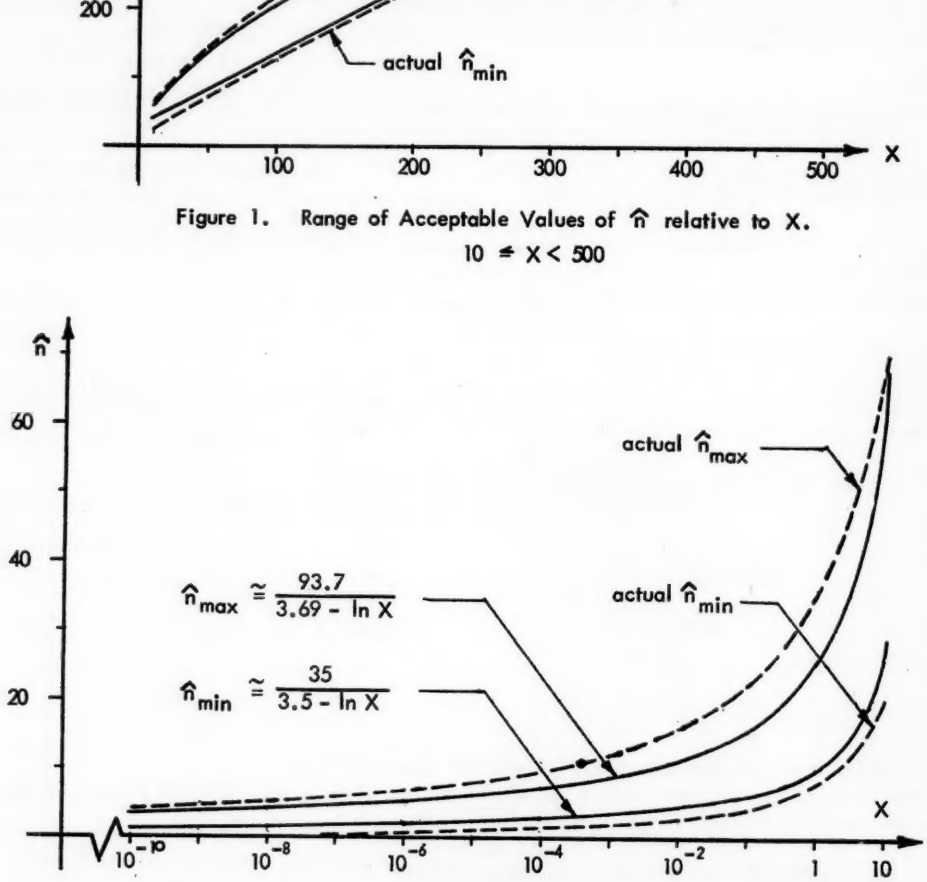


Figure 2. Range of Acceptable Values of n relative to X.
Approximations valid for the range $10^{-51} \leq X < 10$

to x, that the process will not yield reliable values of $J_n(x)$. Indeed this is just the situation and this fact provides a means of determining a lower bound ($\hat{n}_{min}$) for $\hat{n}$ relative to x.

An upper bound (n_{max}) for $\hat{n}$ will be forced on the generation of $J_n(x)$ by the restrictions on the size of numbers that may be used. In the experimental work carried out for the determination of these bounds, the IBM 650 with the automatic floating point device was used. Using this system, all numbers must fall within $(10^{-51}, 10^{49})$. It was found that the maximum number size was attained, not from generation of the $Y_n(x)$, but from calculation of the normalizing constant K in equation (3). In addition, of course, $\hat{n}$ will be limited by the amount of storage space available in the particular machine, but this limitation is not considered here.

Proceeding as suggested above, experimental values have been determined for $\hat{n}_{min}$ and $\hat{n}_{max}$ relative to x. In presenting the results graphically it was found convenient to consider separately the two ranges $10^1 \leq x < 500$ and $10^{-51} \leq x < 10^1$, which may be found in figures 1 and 2 respectively. A useful set of approximating curves for $\hat{n}_{min}$ and $\hat{n}_{max}$ are also illustrated in figures 1 and 2. The time necessary to calculate both $J_n(x)$ and $Y_n(x)$ for $n = 0, 1, \cdots, \hat{n}$ is approximately linear with $\hat{n}$. For the IBM 650, time in sec $\sim .13\hat{n}$.

Conclusion

1. It is possible to generate tables of $J_n(x)$ and $Y_n(x)$ by their recursive relations (1) and (6), and obtain reliable results. The generation of $J_n(x)$ is carried out by a "backwards" process using (1), and the $Y_n(x)$ are generated forwards using (6).
2. By using single precision floating point arithmetic (8 digit mantissa and 2 digit characteristic) it is possible to obtain results with a maximum error of 1 in the sixth significant digit. No attempt was made to determine what effect the use of double precision arithmetic would have on the number of correct significant digits obtainable.
3. In general, the accuracy is not improved by selecting an $\hat{n}$ greater than $\hat{n}_{min}$.
4. The choice of $\hat{n}_{max}$ for $\hat{n}$ in generating $J_n(x)$ was sufficient in all cases to guarantee that $Y_n(x) < 10^{49}$ for $n \leq \hat{n}$.
5. This technique, though fast, is best used when $J_n(x)$ and $Y_n(x)$ are desired for an entire range of n, rather than for a specific n.

A SUBROUTINE METHOD FOR CALCULATING LOGARITHMS

R. W. BEMER, I. B. M. Corporation, 590 Madison Avenue, New York 22, N. Y.

This note describes a method for calculating logarithms which is particularly suitable to computers with Variable Execution Time operations (VET). It was first used for the IBM 705 in the PRINT I and Autocoder systems and is currently being adapted for the IBM 709. The basic method is adaptable for computers which do not have VET operations but may not be competitive with other methods in these cases. The method is described for use with floating point arguments, but it may be adapted, with some preliminary modifications, for computation of logarithms of fixed point arguments.

It is desired to compute $\log_b N$, where the argument or number N is represented in floating point as:

$$N = M \cdot b^E$$ where the exponent E is an integer and

(decimal) $.1 \leq M < 1.$

(binary) $.5 \leq M < 1.$

Then, $\log_b N = E + \log_b M$

The basic principle of this method is cyclic multiplication of the mantissa M by one or a series of specially chosen multipliers, as necessary, until the resultant product falls in a specially chosen range starting at 1.0. A relaxed Taylor series for the log $(1 + x)$ is then used, represented here as $P_s(x)$, where $_s$ represents the number of digits of accuracy in decimal. Let the final product in the range be m_k, so that:

$$1 \leq m_k \leq (1 + \Delta), \text{ where } m_i = M \prod_{i=1}^{k} A_i$$

$$\text{Then, } \log_b M = \log_b m_k - \sum_{i=1}^{k} \log_b A_i$$

Various considerations of the number base and machine characteristics dictate the value of Δ and the multipliers chosen. The multiplier A_i is selected from a table, one entry for each interval of the value of m_i. It is chosen such that when multiplied by the largest value of m in each interval, the resultant product does not exceed $1 + \Delta$. For both binary and decimal methods outlined here, either 1, 2 or 3 multiplications may be required before the resultant product exceeds 1. The average number of times thru the loop is about 1.7 for both methods.

Decimal Machines

The interval chosen here was on the basis of the first significant digit of the running product. Δ was chosen as .1 (Δ must not exceed the maximum interval of argument). Multipliers were chosen with a minimum number of digits for faster VET computation.

First Digit of Running Product	Range of Argument	Multiplier	$\log_{10}$ of Multiplier
.1	.1–.2	5.5	.74036 26894 94
.2	.2–.3	3.6	.55630 25007 67
.3	.3–.4	2.7	.43136 37641 59
.4	.4–.5	2.2	.34242 26808 22
.5	.5–.6	1.8	.25527 25051 03
.6	.6–.7	1.5	.17609 12590 56
.7	.7–.8	1.3	.11394 33523 07
.8	.8–.9	1.2	.07918 12460 48
.9	.9–1.0	1.1	.04139 26851 58

Approximating polynomials $P_n(x)$ for the $\log_{10}(1 + x)$ are produced from the Taylor series by Tchebysheff relaxation of the higher order terms and further relaxation to minimize the number of digits in the highest order remaining coefficient so the first multiplication in the polynomial expansion is VET fast. In the range $0 \leq x \leq .1$:

For 8-digit accuracy (max error = 32 in 10th decimal position),

$$\log_{10}(1 + x) \cong P_8(x) = .4342\ 9394\ x - .2170\ 981\ x^2 + .14327\ x^3 - .09\ x^4$$

For 10-digit accuracy (max error = 68 in 12th decimal position),

$$\log_{10}(1 + x) \cong P_{10}(x) = .43429\ 44627\ x - .21714\ 4958\ x^2 + .14466\ 55\ x^3 - .1066\ x^4 + .0683\ x^5$$

Individual variations may be constructed on these lines. For example, a smaller range might be chosen for $P_8(x)$ so that the 4th order term would be completely eliminated within permissible error limits. This would in turn dictate more and smaller increments in the table.

Example: Find $\log_{10}(36)$

$= \log_{10}(.36 \cdot 10^2)$

.36			
2.7	(A_1)	$-\log A_1 =$	$-.43136\ 37642$
.972			
1.1	(A_2)	$-\log A_2 =$	$-.04139\ 26852$
1.0692		$P_8(.0692) =$	$.02905\ 89481$
		$E =$	2.
	Summing,	$\log_{10}(36) \cong$	$1.55630\ 24987$

Note: The values of A_i must be used as multipliers, not multiplicands, else the VET advantage will be lost.

Binary Machines

The most feasible and general method of logarithm computation for binary computers seems to be to compute $\log_2$ initially and then convert to either $\log_e$ or $\log_{10}$ by a single end multiplication in floating point. Since in binary floating point

$$.5_{\text{decimal}} \leq M = .1\boxed{xxx}xxxx \ldots < 1.0_{\text{decimal}},$$

an 8-position table was chosen to represent the 3 enclosed bits above. Address formation may thus be done by extraction or some other direct method. The basic consideration in the determination of a value for Δ was the elimination of the 4th order term by Tchebysheff relaxation, still maintaining an error less than 1 in the 27th bit for the $\log_e$. (This was chosen for the 27 bit mantissa of a floating point number of the 709 and may be varied for other machines.) $\Delta = .0390625_{\text{dec}}$ was chosen for the table below, being the largest 6 bit binary fraction less than the Δ which meets the above specifications.

Extracted Bits	Range (octal)	Multiplier (octal)	Log_2 of Multiplier (octal)
000	.40–.44	1.66	.703 723 450 561
001	.44–.50	1.52	.564 543 765 447
010	.50–.54	1.40	.453 400 321 640
011	.54–.60	1.30	.353 165 174 015
100	.60–.64	1.20	.244 647 411 363
101	.64–.70	1.14	.176 740 553 440
110	.70–.74	1.06	.102 142 610 633
111	.74–1.0	1.02	.026 565 575 654

In the range $0 \leq x \leq .0390625$, for 27 bit accuracy (max error = 1 in 27th bit for $\log_e$, .5 in 27th bit for $\log_{10}$):

$$\log_2 (1 + x) \cong P_8(x) = 1.342\ 520\ 342\ x - .560\ 7625\ x^2 + .35\ x^3 \text{ (octal coeff.)}$$

Example: Find $\log_e (110.1101)$

$= \log_e (.1101101 \cdot 2^3)$

In octal,

$$\begin{array}{r} .664 \\ 1.14 \quad (A_1) \\ \hline 1.0056 \end{array}$$

$$\begin{array}{rl} -\log A_1 = & -.176\ 740\ 553\ 440 \\ P_8(.0056) = & .010\ 177\ 475\ 366 \\ E = & 3. \\ \hline \log_2 (110.1101) = & 2.611\ 236\ 721\ 726 \end{array}$$

To verify, $\log_2 (110.1101) = 2.76818\ 43186_{\text{dec}}$

$\log_e (110.1101) = 1.91875\ 916$

GENERAL PURPOSE PROGRAMMING SYSTEMS*

ANATOL W. HOLT, UNIVAC Applications Research Center, Philadelphia, Pa.

In these remarks, I do not propose to describe an automatic coding system. My aim is, rather, to present an approach to automatic coding or, if you prefer, to present a motivation which has guided us in the development of the three actual systems for Remington Rand computers. The earliest of these three systems, developed for the UNIVAC I, has been known by the name Generalized Programming. (This system was recently renamed FLEXIMATIC by the Remington Rand Sales Office.) In operation for approximately two years, GP is currently employed at a considerable number of commercial installations. The second system, Generalized Programming Extended (GPX), is a new version of FLEXIMATIC for the UNIVAC II; it has considerably increased powers in comparison with its ancestor.

* Based on a talk delivered at the ACM conference in Houston, Texas, June 19–21, 1957.

Currently, we are building the third system for the LARC computer. During years of development, the aims we wish to achieve have only gradually become clarified. Therefore, the system now in preparation for the LARC represents the fullest realization so far of these aims. The ensuing discussion will use the term "the system" to refer to an embodiment of the aims to be described.

In point of broad classification, the automatic programming system developed by the UNIVAC Applications Research Center is a compiler. This means that it shares two important characteristics with other systems similarly classified. First, complete solution of a problem with the aid of the system involves two computer steps: (1) translation of an encoding of the problem (in a special form associated with the system) into a computer-coded routine, and (2) actual problem computation. The second characteristic common to all compilers is that during the translating process, reference is made to a collection of packages of information, usually called library subroutines, which contribute in some way to the final result.

The system differs from other compilers, however, in that it aims to provide the programmer with a set of services equally applicable to the solution of any problem—services, in other words, which are claimed to be altogether "general purpose." Therefore, the system contains no special design features which make it better adapted to one area of computation than to another.

This aim, as stated, raises a peculiar question: A general purpose computer is, itself, supposed to provide a code which is equally well adapted to all areas of computation. To claim that it is profitable to build a programming system which contributes equally to all problems is, of course, to claim that some facilities which have "general purpose" standing are not realized in the hardware of a general purpose computer. We believe that such unrealized facilities exist and that they lay the basis for a powerful programming system. The facilities in question are an addition to the computer and, in no sense, a subtraction. Hence, a program in the system code may possibly be a one-to-one image of a computer-coded program.

The general purpose facilities I have in mind fall into two broad categories, the first of which—naming aids used in writing computer instructions—has already been widely recognized in the computer field. Naming aids include symbolic addressing, convenient abbreviations to represent computer instructions, special conveniences in the naming of constants, etc. Such naming aids, while interesting in their own right, have already been exploited to a greater or lesser extent in many existing assembly routines. It is apparent that these conveniences are equally applicable to all programs in which computer instructions must be used.

The second facility for which I claim general purpose standing is the ability to use mnemonic and abbreviated notations to represent computations of some given type. In part, this means the ability to use, in addition to the elementary computer instructions, special purpose instructions which are helpful in a given type of computation. By a few examples, I hope to show you the range of special notations I have in mind.

Suppose, for instance, that I have frequent occasion for evaluating sines of angles and that the computer does not contain a special instruction for sine evaluation. In addition to my usual command vocabulary, which the computer interprets directly, it would be a considerable convenience to write a special command which represents sine computation. Perhaps in different problems I require sines with different degrees of accuracy. In that case, my notation for sine computation may also mention a desired accuracy.

In any event, the special notations which represent the sine computation may be viewed as a special encoding of the sine function which must be translated into computer code before the program may function. Such translation is normally accomplished by reference to a so-called library subroutine, introduced by a compiler at the point at which the special notations were originally written. Sine computation, with or without parameters, exemplifies a large family of cases in which it seems convenient to introduce special notations to represent functions which the computer cannot execute directly.

Another example of special notations not usually considered to have much in common with a sine routine call is the use of algebraic notation to represent the evaluation of algebraic expressions. A little reflection shows, however, that both of these cases have important properties in common. In fact, they are both special notations used as part of coding to represent certain types of computations. In the case of sine with specifiable accuracy, we "code" a sine routine by mentioning the word sine and a chosen

accuracy number. In the case of algebra, we "code" an algebraic evaluation by writing the appropriate expression with algebraic notation. In both cases, these special notations must combine with information stored elsewhere before they become coding which is intelligible to the computer.

In the case of special notations for sine, the "other information" is usually thought of as a library subroutine. A compiler then serves to interpret the special notations by finding the library subroutine and, perhaps, by inserting a specified accuracy value into the library coding. In the case of algebra, it is common to think of the "other information" as an algebraic translator which operates as an entity by itself. It is my contention that *either* special notations for algebra *or* special notations for sine may both be handled by a general purpose compiler which has broad ability to combine library information with special notations employed in the original coding.

Still other types of special notations which may offer great conveniences in the writing of code for special problems may also be handled by a general purpose compiler. For example, suppose that I have a file of information on which several different types of processings are to be accomplished. This file contains complex items, each of which is subdivided into many fields containing quantities that may be mnemonically named. It would now be a great convenience if, in my coding, I could refer to the quantities contained in this item by mnemonic names which have been agreed upon as appropriate to that file of information.

A general purpose compiler can assist the translation of such special mnemonics with appropriate packages of library information. A package of library information is combined with special notations employed in the original coding to produce coding in another form. The combining agent is a general purpose compiler. The library packages discussed here have wide use and diversity of structure; so-called library subroutines, on the other hand, are only one example of the library packages which a general purpose compiler can use.

In the development of most compiling systems, the designers begin with a special range of problems for which they intend their system to be useful. They invent special notations which they consider naturally adapted to the area of problems in question, and then they construct a translating mechanism with just enough generality to cope with these notations. While a given set of special notations for a given area of problems may, indeed, be very useful and important, the ability to add arbitrary, *new* notations is even more important than any given set of special notations.

The truth of this may be readily appreciated if we realize that no one set of special notations can be optimally adapted to all ranges of problems. No single computer would be optimally adapted to all types of computation. Furthermore, we cannot even hope, once and for all, to subdivide the entire universe of problems into fixed areas, each with a well-adapted notation. For example, if I have devised one notation satisfactory for business problems and another notation satisfactory for algebra, I am still unable to write convenient coding for problems which lie squarely between these two areas—namely, those problems which involve a lot of data handling *and* algebraic computation. Even if problems which cross the boundary lines never arose, the inability to add new notations would, nevertheless, be hampering. Suppose, for example, that a notation has been developed to handle all algebraic problems with equal facility. If, now, in a certain set of problems, I use frequently either a particular algebraic function or a particular succession of steps, I will find that the general algebraic language at my command causes me to write much more than the particular set of problems requires. In short, as soon as the range narrows from all of algebra to a special sub-domain of algebra, the general algebraic notation ceases to be well-adapted.

In summary, it is possible to construct a general purpose compiler which (a) provides special conveniences in the writing of computer code (naming aids) and (b) provides the ability to add new notations to my repertoire for instruction writing whenever such notations seem profitable for particular ranges of problems. Of course, profit is a relative thing. In exchange for the advantages bought by having certain new notations available, I must labor to add to the library packages which make translation of these notations possible. Consequently, this central feature of the general purpose compiler is truly meaningful only if the labor required to obtain the use of new notations is comparatively small.

The theoretical foundations for such compiling systems are now being developed at Remington Rand. Early examples, such as GP for UNIVAC I, demonstrate the practical utility of a system designed from this point of view.

AN IMPROVED DECIMAL REDUNDANCY CHECK

ROGER L. SISSON, Canning, Sisson & Associates, Los Angeles, Calif.

As more emphasis is placed on improving the accuracy of data fed into automatic computing systems, more emphasis will be placed on redundancy checking of predicable fields within the input.

Two systems (at least) of checking a field of decimal digits have been proposed[1, 2]. In both of these it is assumed that the field to be checked is all numeric and that the redundancy must be of only one digit. These assumptions will be retained in this note.

Generally the check is made in this way: When the field (e.g., shop order number) is assigned, it is assigned with the check digit appended. The check digit is either computed at the time of printing or is read from a table of available numbers.

When the data is fed back into this system all fields which were pre-assigned are checked by recomputing the check digit and comparing it with check digit entered.

Let the field to be checked be composed of decimal digits:

$$d_n, d_{n-1}, \ldots d_i, \ldots d_1, d_c,$$

d_c being the check digit which is appended. (i = the power of the base, 10.)

In one method[1], d_c is formed thus

$$(1) \qquad d_c = \left[\sum_{\text{i odd}} d_i + \sum_{\text{i even}} 3\,d_i \right]_{\text{mod } 10}$$

(The system may be mechanized with the 3 before or after the summation.)

In the second method[2] the check digit, d_c, is formed thus:

$$(2) \qquad d_c = 10 - \left[\sum_{\text{i odd}} \left(2d_i + \frac{2d_i - [2d_i]}{10} \right) + \sum_{\text{i even}} d_i \right]$$

The most common clerical errors are either to copy a wrong digit or to transpose adjacent digits. The above schemes do not detect *all* transpositions. In (1), if two adjacent digits d_{k+1}, d_k differ by 5, then the check fails when the digits are transposed since[4]:

$d_c = [d_{k+1} + 3d_k + A]$ and transposed

$d_{c'} = [d_k + 3d_{k+1} + A]$ where A = other terms which are equal in both cases.

If $d_{k+1} = d_k \pm 5$ then

$d_c = [d_k \pm 5 + 3d_k + A]$ and

$d_{c'} = [d_k + 3d_k \pm 15 + A] = [d_k + 3d_k \pm 5 + A] = d_c$

Thus this type of error is not detectable. (The situation is symmetric and so applies also if d_k is an odd digit.)

It can be shown that (2) fails if the adjacent digits are 0, 9.

No check system, in fact, restricted to using 10 symbols can check *all* adjacent transpositions. However, if a base of eleven or more is used such checks are possible[4]. Since in actual use in a computing system numbers are usually coded in binary form, the base of the system is actually 16 (or 64 in alphanumeric cases). An example is given of how this fact may be used to improve the checking. We can make the first check system, (1), detect all adjacent transpositions if we take advantage of the fact that soon after the decimal number is inserted into an input device it is converted to a binary-coded decimal form.

The check digit is computed in the base 16:

$$d_c = \left[\sum_{\text{i odd}} d_i + \sum_{\text{i even}} y \right]_{\text{mod } 16}$$

where $y = 3d_i$ if $d_i \leq 7$

$y = 3d_i - 1$ if $d_i > 7$

---

[1] Canning, R. G., *Electronic Data Processing for Business and Industry*, Wiley, 1956, p. 189–191.

[2] "Preliminary Manual for Self-Checking Number Device for Types 24-26 Card Punches," International Business Machines Corp., N. Y., N. Y., form 22-6022-0; p. 4–5.

[3] "Preliminary Manual of Operation, IBM, EDPM Type 705," International Business Machines Corp., 590 Madison Ave., New York 22, N. Y., Form 22-6627-3, p. 120.

[4] First brought to our attention by a private communication from Wm. A. Donaldson, Rolls-Royce Ltd., Glasgow, Scotland.

The logical equations to compute y are not hard to write, see appendix. To show that all adjacent transposition errors are detected, we ask:

$$d_{k+1} + y\,d_k \stackrel{?}{=} d_k + y\,d_{k+1}$$

Let $d_{k+1} = d_k + \Delta; \qquad \Delta > 0$

Then $d_k + \Delta + y\,d_k \stackrel{?}{=} d_k + y\,d_k + y\,\Delta$

or $\Delta \stackrel{?}{=} y\,\Delta$

if Δ =	then $y\,\Delta$ =	or
1	0011	3
2	0110	6
3	1001	9
4	1100	w
5	1111	z
6	0010	2
7	0101	5
8	0111	7
9	1010	u

Thus $\Delta \neq y\,\Delta$ (mod 16) and all such errors are detected. Since all 9 symbols of $y\,\Delta$ are different, all wrong digits are also detected. There are two restrictions to observe:

1. d_c must be computed as a binary sum.
2. d_c may turn out to be an alphabetic character or symbol, which must be acceptable to the computer.

In regard to 2, if the check is made at the input device, as is most desirable, the computer need not even accept the check character. It could automatically be deleted after the check by the input device (it being assumed that other checks, e.g., parity, are then used.) Generally, the computer system *will* have to be able to store and print all check characters in order to create documents containing fields with the check included. This may be difficult if the characters happen to represent end-of-tape, blanks, etc. However, d_c can be made to assume usable characters with some simple logical manipulations.

For example, for the 705 code[3] the following rules might be used to transform d_c to a usable character. (The parity bit would be appended in the usual way to d_c.)

1. Make the following transfers:

binary positions of d_c		*705 code binary positions*
1	⟶	B (always a "one")
b_4	⟶	A
0	⟶	8
b_3	⟶	4
b_2	⟶	2
b_1	⟶	1

2. Add binary one to the resulting number.

With these rules d_c will be A thru H or J thru Q only, all legitimate symbols.

If it is important to catch *all single* adjacent transpositions, the logical circuits added to the input device to accomplish the above system would not be expensive. Such assurance is valuable if justification of a system to auditors and management depends upon their being convinced that the system has the proper level of accuracy in spite of the common clerical mistakes.

The idea of taking advantage of the binary-coding of the input digits for check purposes can perhaps be expanded; for example, to detect the common key punch error of striking keys in the wrong horizontal row.

APPENDIX

The logical equations for the binary digits of y (y_1, y_2, y_3, y_4) can be derived from the binary digits of d_i (b_1, b_2, b_3, b_4) by the following equations (not necessarily component minimizing):

$$y_4 = b_4\,\bar{b}_1 + \bar{b}_4\,b_1$$
$$y_3 = (\bar{b}_3\,b_4 + b_3\,\bar{b}_4)\,\bar{b}_1$$
$$y_2 = \bar{b}_1\,(\bar{b}_2\,b_3\,\bar{b}_4 + b_2\,b_4 + b_2\,\bar{b}_3) + b_1\,\bar{b}_2\,\bar{b}_3\,\bar{b}_4$$
$$y_1 = \bar{b}_1\,b_4\,(\bar{b}_2\,b_3 + b_2\,\bar{b}_3) + \bar{b}_3\,(\bar{b}_1\,b_2\,\bar{b}_4 + b_1\,\bar{b}_2\,b_4)$$

(Sub 4 is least significant digit).

BINARY AND TRUTH-FUNCTIONAL OPERATIONS ON A DECIMAL COMPUTER WITH AN EXTRACT COMMAND

WILLIAM H. KAUTZ, Stanford Research Institute, Palo Alto, California

It occasionally becomes desirable to solve, on automatic digital computing machines which are capable of handling only decimal numbers, problems in logic, class structure, coding, binary relations or binary arithmetic. This note describes how the major logical and binary operations can be carried out on one such machine, the DATATRON 205, without any circuit modifications to the computer. These procedures would be applicable with little modification to any decimal computer with an extract command, however.

The EXTRACT command of the DATATRON 205 follows these rules:

$$aEb = \begin{cases} 0 \text{ if } a \text{ is even} \\ a + b - 1 \text{ if } a \text{ is odd} \end{cases}$$ Where a and b are single decimal digits.

In conventional use the digits of the extractor a are decimal 0 and 1 only, so:

$$aEb = \begin{cases} 0 \text{ if } a = 0 \\ b \text{ if } a = 1 \end{cases}$$

A. *Logical operations* (no carries)

1. Number representation: An n-variable truth-functional conjunct, an n-bit code element, and a vertex on the binary n-dimensional cube may all be unambiguously represented by an n-digit binary number. Represent this binary number on the computer as if it were a positive n-digit decimal number—that is, represent binary 0 and 1 by decimal 0 and 1, respectively. For more than ten binary digits, more than one word position is required, but since none of the logical operations below involve a carry, they may be performed successively on the multiple words of the long binary number without multiple precision.

   Let X, Y, . . . etc. indicate binary numbers and x, y, . . . etc. their decimal representations. Let xEy indicate the result of extracting x on y (i.e., x is the extractor and y is the number on which the extraction is being performed). For the 10 decimal digit word, let n* represent nnnnn nnnnn, thus: $1^* = 11111\ 11111$ and $9^* = 99999\ 99999$. Use $\doteq$ to designate "equal by representation", so $X \doteq x$

2. Complementation (NOT): $\overline{X} \doteq 1^* - x$.
3. Conjunction (AND): $XY \doteq xEy = yEx$
4. Inequivalence (Exclusive OR): $\overline{X} \oplus Y \doteq (x+y)E1^*$
   or $X \oplus Y \doteq (x+y)E(x+y)$
5. Disjunction (Inexclusive OR): $X_vY \doteq (x+y) - (xEy)$
6. Biequivalence: $X \equiv Y = \overline{X \oplus Y} \doteq [x+(1^*-y)]$
   $= \overline{\overline{X} \oplus Y} \doteq [(1^*-x)+y]E1^*$
   $= \overline{X \oplus Y} \doteq 1^*-[(x+y)E1^*]$
7. Equality (X = Y?): Form $x - y$ or $X \oplus Y \doteq (x+y)E1^*$, and Branch on non-zero (CNZ or NOR)‡
8. Implication ($X \supset Y$?): Form $X\overline{Y} \doteq xE(1^*-y)$ and branch on non-zero.
9. Distance: To count the number d of 1's in X ($d \leq 9$), form $1^* \cdot x$ and extract the center digit of this product. If numbers are right-justified within the words, this digit is the n-th from the right

‡In this and succeeding decisions, branching of the program constitutes a "no" answer to the question.

end of the R-register. If the numbers are left-justified, this digit is the (n + 1)st from the left end of the A-register. If there is a possibility that d = 10, test for this possibility separately (by forming $\bar{X} \doteq 1^* - x$ and branching on zero before the multiplication). Example for n = 5:

```
x   = 10110
1*  = 11111
      -----
      10110
     10110
    10110
   10110
  10110
  -------
1*·x = 112332210
       ↑_________d = 3
```

10. Unit distance: To determine if two numbers X and Y differ in only a single bit position, form $X \oplus Y \doteq (x + y)E1^*$, normalize, subtract the number (10000 00000), and branch on non-zero. The digit position (less 1) in which X and Y differ appears in the special counter. If X = Y, the normalize operation itself causes a branching, thereby achieving a full trichotomy (d = 0, d = 1, d > 1).

11. Distance = d?: To obtain a multiple branch on the basis of the number d of differing bit positions of two numbers X and Y, carry out (10) above, but repeat the normalize and subtract operations d times. Alternatively, add the distance d determined by (9) above to a relocating address in the next command.

12. Even parity?: To determine if a (right-justified) number X has an even number of 1's, form 1*·x, clear the A-register, shift left one digit, extract the result on 1*, and branch on non-zero.

B. *Binary arithmetic* (with carries)

1. Number representation: Here let binary 0 and 1 be represented by decimal 0 and 9, respectively. For n > 10, multiple precision arithmetic is required.

2. Addition and subtraction: $X \pm Y \doteq 9^* - [\{9^* - [(x \pm y)E1^*]\ E1^*]$

Example:

```
      101001 ⟶          909009
Add   001011 ⟶    Add   009099
      ------            ------
      110100            918108
                  Extract on 111111
                             ------
                             910100
                Subtract from 999999
                              ------
                              089899
                  Extract on 111111
                             ------
                             009099
                Subtract from 999999
                              ------
      110100 ⟵          ⟵  990900
```

Note that the last 4 steps are required to convert 8's to 0's and 1's to 9's. For simple counting (X + 1), the last 3 steps (subtract-extract-subtract) may be omitted.

3. Multiplication, XY: Perform by repeated addition of the multiplicand X under control of the multiplier (Y ÷ 9). The steps of this process are familiar and standard, except that the "addition" step is that of (2) above and not a simple Add command.

4. Division, square root, etc. can be performed by well-known techniques in terms of the "addition" and "subtraction" of (2) above.

The author is indebted to Ralph Keirstead of Stanford Research Institute for his collaboration in the development of the above techniques.

UNUSUAL APPLICATIONS DEPARTMENT

AUTOMATIC IMPLEMENTATION OF COMPUTER LOGIC

By E. F. Morris and T. E. Wohr
Product Development Laboratory, I.B.M. Corp., Poughkeepsie, N. Y.

INTRODUCTION: Implementation is the process of fitting specific circuits to computer logic. This involves carrying out the calculations necessary to insure proper current drive, consistent with the time required to perform the logic. Hand methods of implementation are considered impractical because of the time involved in going through the tedious calculations necessary to fit the transistor circuits into the chains of logic in a computer. These lengthy calculations are required chiefly because a given set of circuits must be operated at an optimum speed to perform the logic in a specified time and also maintain a high degree of reliability.

This program, carried out on the IBM 650 Data Processing Machine, has been used very successfully at the IBM Poughkeepsie Laboratory to implement computer logic. Results that would take a month to reach by hand methods can be produced in a single day using the machine program. Equally important is the fact that the large amount of error inherent in manual implementation has been minimized in the machine program. For example: if the hand method were used, it would be necessary to take a good deal of the information from curves like those in Figs. 1, 2, and 3, and errors would result from inaccurate interpolations. In the machine program, these and other curves have been fitted to formulas, and are stored as such in the machine.

TRANSISTOR LOGIC: At this point in the discussion, to provide a clearer picture of the implementation process, it is advisable to supply some background on the use of transistors in performing logic. One of the most fundamental jobs of a computer is to perform these logical operations. The most basic are those of the "and" and "or" functions. In executing these operations, we could make use of any one of several nonlinear circuit elements. However, we will confine our attention to the use of the transistor in logical configurations, because it approximates an ideal switch, and also because of the present emphasis on speed and space requirements in digital computers.

The Boolean expression for the "and" function is written $x \cdot y$. This means that both the x and the y inputs must be "one" to enable the output to be "one." In other words, the output will follow the most negative input if we consider a "one" to be the positive or "up" level input and zero to be the negative or "down" level input. As shown in Fig. 4, two PNP emitter followers driving a common load resistance R_e will produce the "and" operation. If the emitter is positive relative to the base, the base has control and the output will be clamped at approximately the base level. However, if the emitter to base voltage is negative, the device is cut off and the input will have no control. Thus, if two or more PNP emitter followers are placed in parallel, the output will be controlled by the most negative input.

The NPN transistor complements the action of the PNP unit to produce the "or" function, $x + y$, as shown in Fig. 4. In this case, if the emitter to base voltage is positive, the transistor will be cut off and will have little effect on the output. If a positive signal is applied to the base, the transistor will conduct, clamping the emitter to very nearly the same positive base voltage. Therefore, if two or more NPN emitter followers are placed in parallel, the output will be controlled by the most positive input, producing the "or" function.

In addition to these two circuits, a buffer or driver circuit is necessary in a string of logic wherever loading conditions demand one. The complementary emitter follower shown in Fig. 5 serves this purpose and is included in the program.

IMPLEMENTATION OF A SIMPLE LOGICAL CHAIN: We now have two of the basic logical circuits contained in the design manual. Assuming that we have also stored their operating specifications in the machine, we can proceed with the implementation of a logical chain.

As an example, we will consider the simplified case of finding the input requirements to the "and" circuit of Fig. 4. It must be kept in mind that in implementing emitter follower logic it is necessary to start from a given load or output and work backwards to find the input parameters. Consequently, the following information is fed into the machine to specify the "or" circuit of Fig. 4.

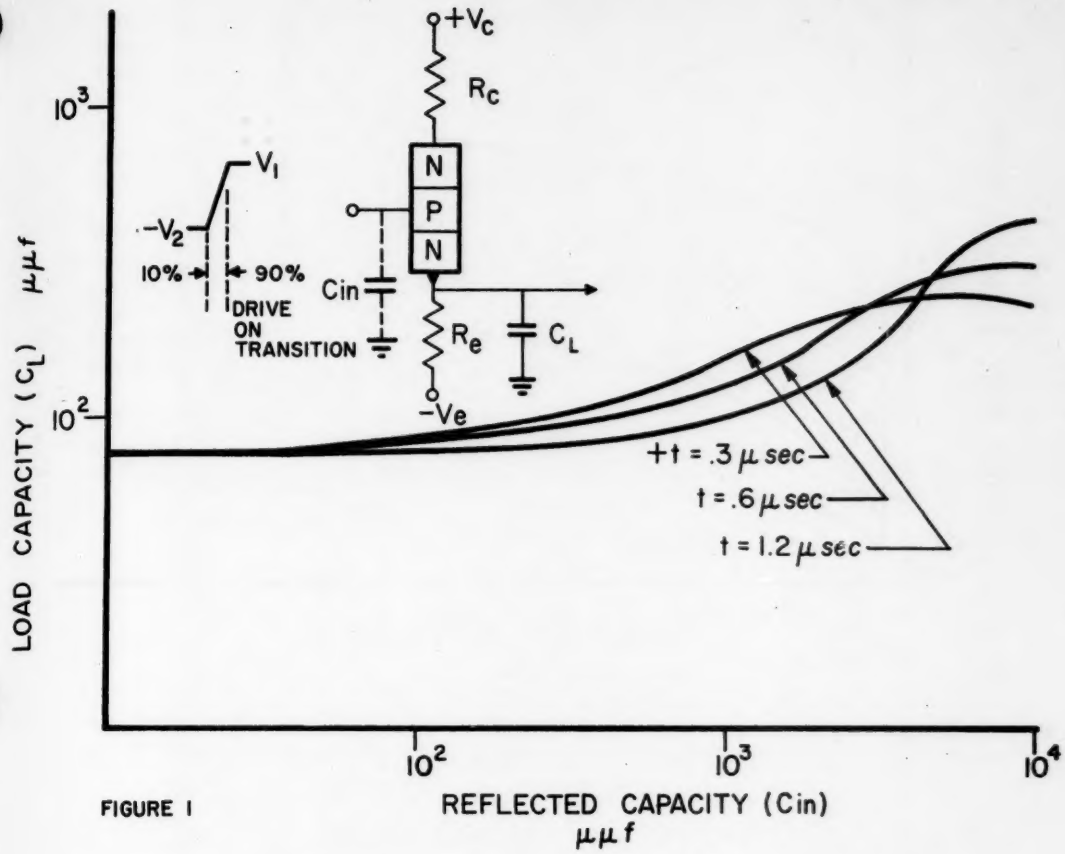


FIGURE 1

a. The load current at both voltage levels.
b. The number of inputs to the stage.
c. Maximum and minimum load capacity (C_L).
d. Maximum allowable turn-off transition time (t_{spec}).

Given this information, the program operates on it to produce the following:

1. The value of the emitter resistance R_e is calculated, taking the tolerances on the resistances and voltages in the worst direction. The program then performs a table look-up routine from which the next lower RETMA value of R_e is found.
2. The maximum allowable load or emitter current dictated by maximum power considerations is calculated, using the formula
$$P_{max} = (E_{eq} - 1.05\ V_e)^2/4\ (E_{eq}/I_{e\ max} + .95\ R_e)$$
and solving for $I_{e\ max}$. This value is then tested to see that it is not exceeded by the actual current flowing through R_e.
3. The maximum turn-off transition time is next calculated from the relationship
$$e/E = 1 - \exp(-t/RC).$$
Solving for t,
$$t_{max} = -R_{RETMA}\left[C \ln\left(1 - \frac{e}{E}\right)\right]$$

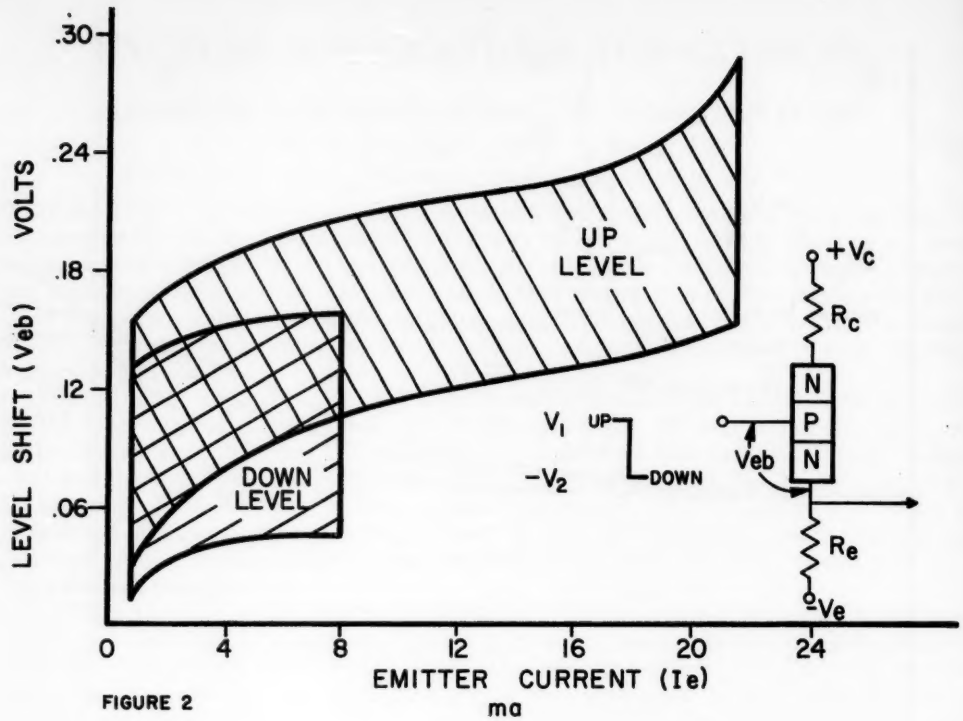


FIGURE 2

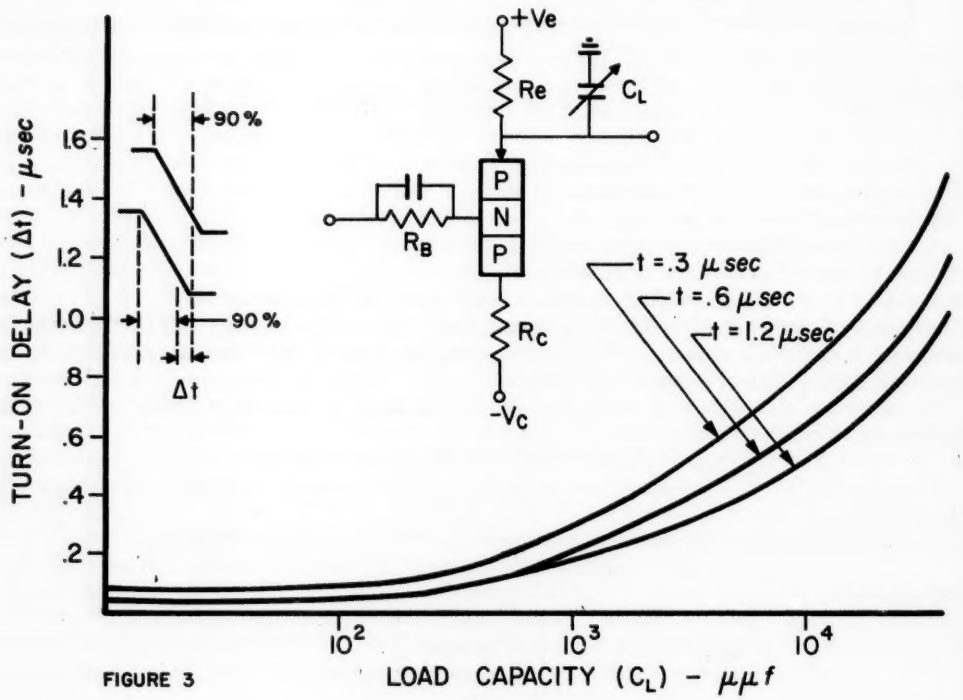


FIGURE 3

The program compares this value of t_{max} with the specified maximum allowable time for that particular stage. If this maximum allowable time is exceeded, an iteration takes place. In it the RETMA value of R_c is lowered until t_{max} is within the maximum allowable time. While this iteration is going on, the program is continually testing to see that the maximum power condition in (2) is still satisfied.

4. The input or base current in the "on" case is then found by dividing the emitter current by α'. The input current in the "off" case (I_{co}) is calculated from the formula

$$I_{in} = I_{co} \times 10^{PK/x}$$

where P is the maximum power found by using the relationship in (2), and K and x are constants.

5. Next the voltage level shift encountered in going through the device (Fig. 2) and the reflected capacity (Fig. 1) are calculated from formulas which provide the best fit to the empirical curves in Figs. 1 and 2. For the voltage level shift curves, several sets of simultaneous equations were solved to produce a polynomial for each of the four curves in Fig. 2. Choosing three points on each curve, the equations took the form

$$V_{eb} = -k_1 I_e^2 + k_2 I_e + k_3$$

This V_{eb} is added to the drop across the base resistance R_B to give the total voltage shift encountered in going through that stage. The family of curves for reflected capacity were a little more difficult to fit because three variables were involved—C_L, t, and C_{in}. The formula

$$C_{in} = \frac{1}{\frac{1}{k_1 + k_2 t^2} + \frac{1}{k_3 t + k_4 C_L}}$$

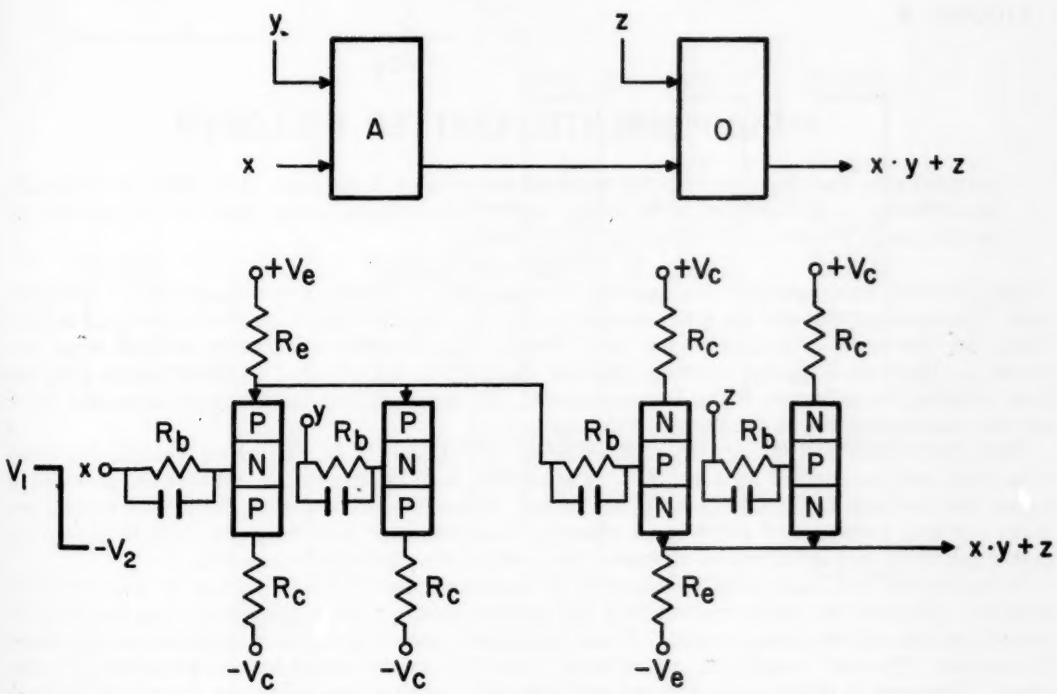


FIGURE 4 "AND" CIRCUIT DRIVING AN "OR" CIRCUIT

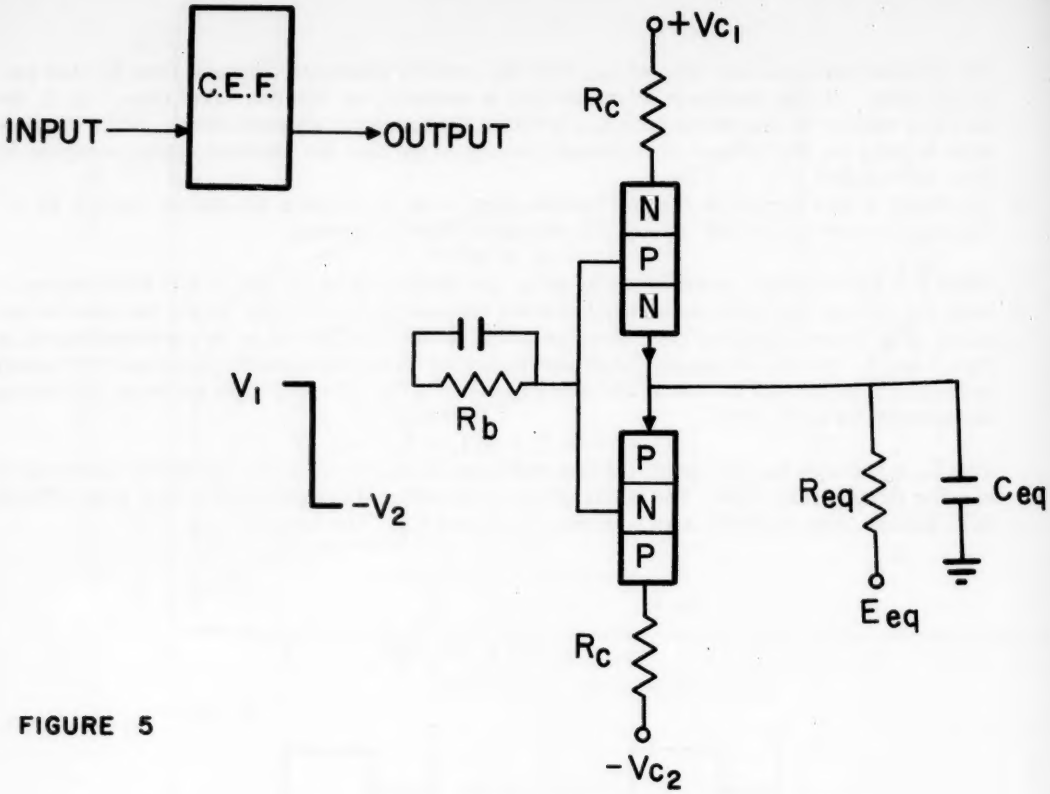


FIGURE 5

COMPLEMENTED EMITTER FOLLOWER

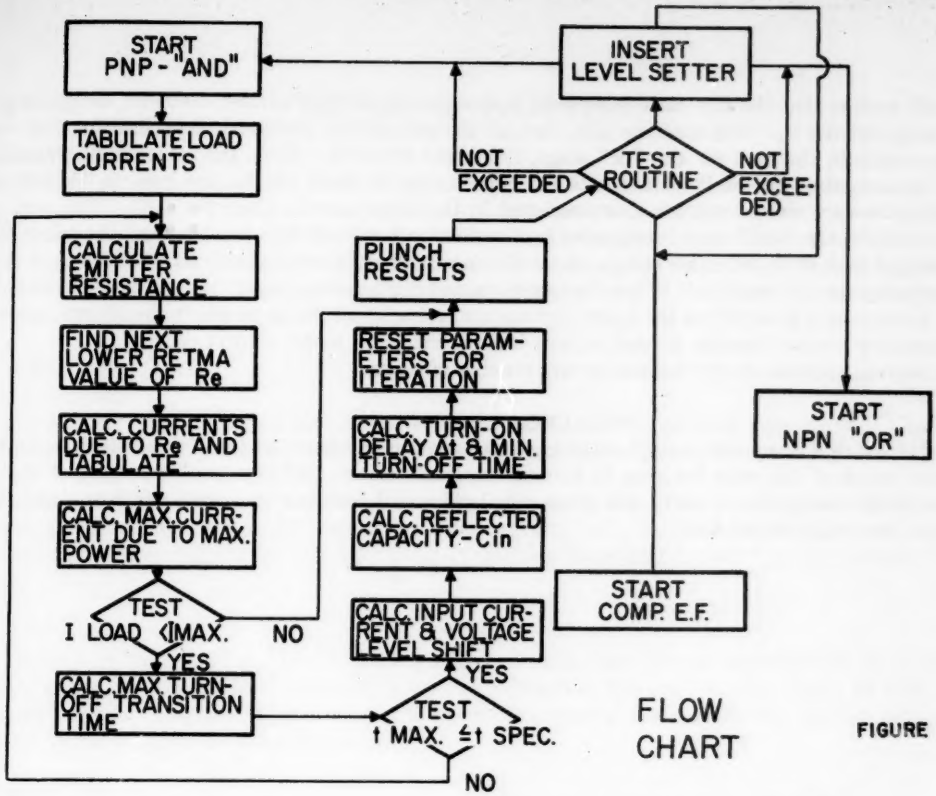
provided a fit that deviated from the empirical curve by ± 2 per cent. The delay (Δ t) through the transistor is also specified in the design manual by empirical curves (Fig. 3). An excellent fit for this family of curves is provided by the relationship

$$\Delta t = (k_1 - k_2 \log k_3 t)\, C_L + k_4$$

The results of these operations can be punched out or printed, whichever is more convenient and practical. The input currents and the input capacity to the "or" circuit of Fig. 4 (as found in 4 and 5 respectively) are then used as the load on the "and" circuit. The preceding calculations for that circuit are repeated. However, if the test routine in the flow chart of Fig. 6 shows that a certain voltage level has been exceeded, an indication of this is punched out. The input currents and capacity to a level setter are then used as the load on the "and" circuit.

IMPLEMENTATION OF A COMPLEX CHAIN: CODING OF LOGIC: Now consider the somewhat more complex chain of logic in Fig. 7. The problem is to implement the entire chain, provided of course that the loads L_1, L_2, L_3, L_4, and L_5 are known. When this has been done, the emitter resistances, input currents, voltage level shifts, input capacity, transition time, and delay time will be known for each stage, along with an indication of where a level setting circuit should be inserted.

To accomplish this, each stage is designated by a control number, which consists of three two-digit numbers. The first two digits (reading from left to right) specify a major control, the next two a minor control, and the last two a stage control. In the 650 program each stage in the chain is completely defined by one card. The card contains the control number and the parameters that have to be specified for that stage. The order in which these cards are read into the computer determines the order in which each stage is operated on by the program.



FLOW CHART

FIGURE 6

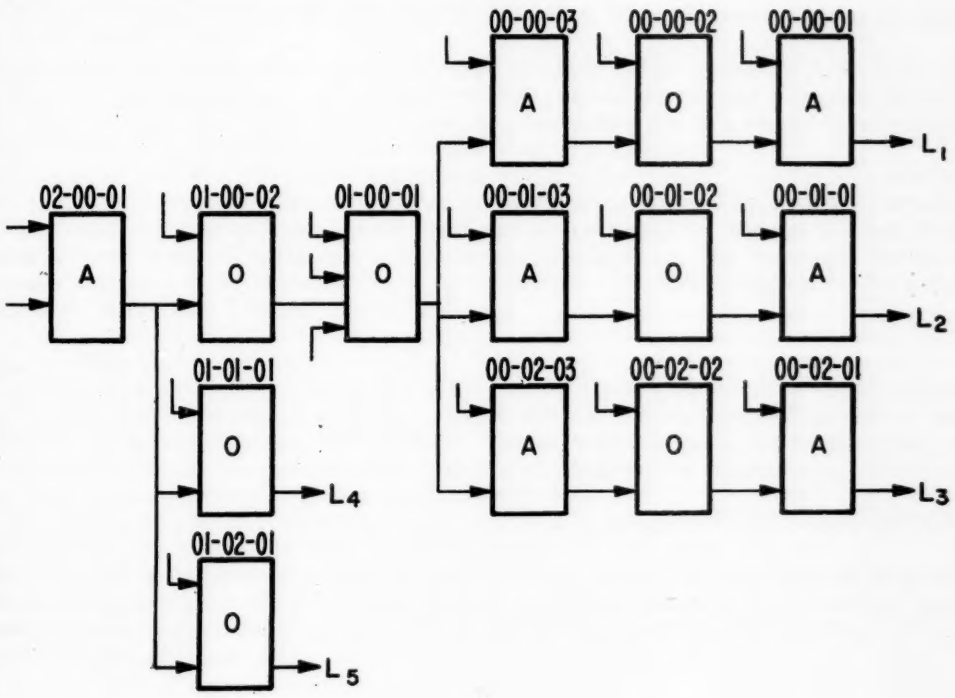


FIGURE 7

We will assume that the first card to be read in contains the control number 00-00-01, designating the "and" stage driving L_1. The machine goes through the calculations previously described for that stage, and then reads in the card for the "or" stage, numbered 00-00-02. Since the major and minor digits remain unchanged at 00, the input currents and capacity for 00-00-01 become the load on 00-00-02, and the calculations are carried out for that stage and in the same way for stage 00-00-03. The next card read in is that of the "and" stage, designated by control number 00-01-01. At this point the minor digits have changed from 00 to 01. This change causes the input currents and capacity for 00-00-03 to be stored in a temporary storage location. When the minor control digits change again, the contents of temporary storage location will be added to the input currents and capacity of the stage where the change occurred. The temporary storage location is reset to zero when a change in major control occurs.

The program operates in this fashion on any number of stages.

CONCLUSIONS

Thus far, using a computer to implement logic has saved many hours of tedious calculations, and has eliminated much of the error inherent in manual implementation. It is reasonable to predict that the need for using a computer to carry out these calculations will continue to increase as larger and faster computers are designed and built.

NEWS & NOTICES

NATIONAL ACM

The Roster of Members of the National Association for Computing Machinery, dated January 1, 1958, was distributed in April and for the first time listed members by geographical areas as well as alphabetically. The total membership exceeds 3,000. The roster shows members located in twenty-one foreign countries. Members in the United States are listed in groups by cities, with Los Angeles and New York City vying for the largest resident listing. Also included in the roster is a complete reprint of the Constitution and Bylaws of the Association.

ACM CHAPTER NEWS

Syracuse Chapter:

At the January 31 meeting of the Syracuse Chapter the following officers were elected: Chairman, Dr. Bruce Gilchrist of Syracuse University; Vice Chairman, Mr. Robert Jones of Sylvania Electric; Secretary-Treasurer, Miss Gloria Bauer of IBM.

The speaker for the February 26 meeting was Dr. John Carr, III, of the University of Michigan and President of ACM. On March 26 there was a joint meeting with the Instrument Society of America, which included a presentation on "An Automatic Production Recording System" by Mr. Gordon Young of IBM.

Houston Chapter:

Sir Richard Southwell, M.A., LL.D., D.Sc., D.Eng., F.R.S., who has an international reputation in the field, spoke on "Numerical Computation by the Relaxation Method" at the March 26 meeting of the Houston ACM Chapter. Approximately fifty people attended the meeting and participated in the discussion which followed Sir Southwell's lecture.

San Diego Chapter:

At the April 17 dinner meeting, Mr. John Stroud of the U. S. Navy Electronics Laboratory spoke on "The NTSD Unit Computer."

Membership in the chapter has reached a new high of ninety, almost double the July, 1957, record.

At the March council meeting, it was decided to publish a special annual issue of the San Diego Computer Newsletter which will include 1) the by-laws, 2) a membership list, 3) a resume of current officers and committees, and 4) a description of the activities of the preceding year.

The election of the 1958–1959 officers will be held soon, and the nominations have been reported as follows: Chairman, Florence Oglebay of NEL, Paul Lewis of Astronautics and Ben Ferber of Convair-SD; Vice Chairman, Jack Grams of Ryan, Dorothy Shumway of Convair-SD and Maurice Halstead of NEL; Treasurer, Edward Bates of Convair-SD, Don Parker of Astronautics and Dan Sonheim of Convair-SD; Secretary, Hal Judd of IBM, Dorothea Jirauch of Convair-SD. The Nominating Committee members are Margaret Covher, Jack Rose and Charles Swift.

Los Angeles Chapter:

At the April 2 dinner meeting, Paul Armer of The RAND Corporation described "Salary Structure in the Programming Profession", a statistical analysis which used a mathematical equation of salary as a function of years of experience since Bachelor's Degree, a growth factor, the starting salary, etc., evolved by Cecil Hastings and member of the RAND staff. Eighty-eight organizations, most of which were engaged in scientific computing, furnished data for the survey.

The program for the May meeting will include a report by H. J. Langlie of ALWAC Computer Division of El-Tronics, Inc., on "Automatic Machine Tool Control."

The chapter will be an active participant in the 1958 WJCC Sponsor's Booth. Local ACM members will solicit membership, sell copies of the proceedings of the 1957 new computer symposium, and dispense additional information about ACM. Similar activity will be maintained by representatives of the other sponsoring societies, AIEE and IRE.

On Friday, May 9, the Los Angeles Chapter will again sponsor a fourth day symposium at the WJCC, "Small Automatic Computer and Input-Output Equipment—A Report from the Manufacturers." No WJCC registration is required for attendance and no fee will be charged for admittance.

The bylaws of the chapter have been revised and will be published and distributed to the members.

Nominations for the officers to be elected in June are reported as follows: Chairman, J. D. Madden of System Development Corporation, J. A. Postley of RAND and R. E. Utman of ElectroData; Vice Chairman, B. F. Handy of Litton Industries, J. Slap of ALWAC Division of El-Tronics Inc., and F. V. Wagner of North American Aviation; Secretary, R. A. Mallet of North American Aviation, R. W. Rector of Space Technology Laboratories; Treasurer, R Rigsby of Lockheed Aircraft, T. Sanborn of Douglas El Segundo and J. D. Tupac of RAND.

UNIVERSITY ACTIVITIES AND EDUCATION PROGRAMS

• We continue our list of university computer installations with information received from the Burroughs Corporation dated February 11, 1958. The following have installed Datatron 205 computers:

Installation	*Location*	*Director*
Purdue University	Lafayette, Indiana	Prof. Karl Kossack
California Institute of Tech.	Pasadena, California	Prof. G. D. McCann
University of Chicago	Chicago, Illinois	Dean Walter Bartky
Stanford Research Institute	Menlo Park, Calif.	Dr. Clay L. Perry
University of Dayton	Dayton, Ohio	
The following have installed Burroughs E101 or E102 computers:		
Harvard University	Cambridge, Mass.	Mr. Albert Beaton
University of Pennsylvania	Philadelphia, Penn.	Dr. Carl C. Chambers
University of Missouri	Columbia, Missouri	
Massachusetts Institute of Technology	Cambridge, Mass.	Mrs. Ruth Lyon
University of Detroit	Detroit, Michigan	Prof. G. E. Markle
University of Cincinnati	Cincinnati, Ohio	Mr. Stanley E. Shayne
New York University	New York, New York	Dr. George Radin
University of Rochester	Rochester, New York	Prof. T. A. Keenan

• Corrections to the list of IBM 650 installations which appeared in the February, 1958 issue: the Director of the University of Washington Research Computer Laboratory is Professor David B. Dekker; the Director of the Computer Laboratory at Rensselaer Polytechnic Institute in Troy, New York is Jack Hollingsworth.

• The following information was received from the IBM Corporation dated March 19, 1958: "In order to encourage the use of data processing equipment by colleges and universities for educational and academic research purposes, IBM has in effect an education contribution plan. This plan has placed large and intermediate scale electronic data processing equipment at a substantial reduction of monthly rental on over forty college campuses.

The IBM Educational Contribution Plan is administered by highly qualified special representatives. These representatives aid the university in developing a sound development program for the use of D. P. equipment in their academic work. As their needs grow, it is possible to install larger and more powerful equipment under the plan.

It is hoped that the accessibility of such equipment will foster an educational program on the part of the colleges and universities which will present modern data processing machines as computational tools. Presumably, university graduates will then be prepared to enter industry with a better understanding of how this equipment is applied to their chosen field. Furthermore, it is felt that faculty and graduate student research programs can be materially aided by the availability and use of data processing machines on the campus. Already, outstanding research in chemical engineering, sociology, psychology, system simulation, and management gaming has been accomplished through the use of these machines."

• Miami University is planning to establish a computing center and expects to have an IBM 650 Computer installed by the fall of 1959. In anticipation of this development, the Department of Mathe-

matics plans to offer in 1958–59 a course or courses in numerical analysis pointed toward machine computation. Any syllabi, notes, outlines, or other recommendations which might be useful to them in planning their courses will be appreciated. They may be sent to Professor H. S. Pollard, Chairman, Department of Mathematics, Miami University, Oxford, Ohio.

• An Interdisciplinary Program in the general field of Language Models and Logical Design is offered by the Graduate School at the University of Michigan. The degrees offered are Master of Arts and Doctor of Philosophy. The Program is administered by a continuing committee of the Horace H. Rackham School of Graduate Studies at the University of Michigan.

This program is designed to cover some problems and fields which stand in a unique relation to the physical sciences. On the one hand these problems involve language, and on the other they require knowledge of the results and techniques of mathematics and engineering. The term "language models" refers to representations of artificial as well as natural languages; the area includes codes, symbolic languages of mathematical logic, and instructional languages for computers. The term "logical design" covers the logical structure and synthesis of both digital (discrete) and analog (continuous) computers, as well as automatic systems capable of responding to artificial and natural languages.

This program is directed toward the mechanical and the mechanizable aspects of linguistic behavior. The emphasis is on models for natural languages and for artificial languages which are closely related to natural languages; interrelations among these models are also studied. The program is devoted to fundamental theories and to applications of those theories both in natural and automatic systems. Such areas as the mathematical theory of information and mathematical logic provide the basic theory for the development of language models. In the field of logical design the emphasis is on the theory of automatic systems capable of performing human functions involving symbolization. The following is a sample list of special topics falling within the area: data processing in relation to the fields of psychology and linguistics, theory of automatic computer programming, psycho-physics of communication, automatic speech production and recognition, mechanical translation of languages, theory of neural nets, simulation of brain models on digital computers, and techniques of automation to perform such human functions as learning and reasoning.

The program involves both the use of computers for investigating language and symbolic processes and the development and use of languages for studying computers. Mechanisms are employed as models to test theories and as experimental tools. All students are required to become familiar with the principles and operation of electronic digital computers. Students engaged in experimental work are expected to acquire a knowledge and skill in the use of electronic instrumentation for research purposes and in systematic experimental techniques.

For further information write to the Chairman of the Committee on the Degree Program in Language Models and Logical Design, Professor Gordon Peterson, 178 Frieze Building, University of Michigan, Ann Arbor, Michigan.

• Remington Rand Univac has announced courses of instruction relative to UNIVAC systems, open (without charge) to university faculty and administrative officials. These courses will be held during the summer of 1958 according to the following schedule:

Business Oriented Courses

Babson Institute of Business, Wellesley, Massachusetts	June 23–July 18
University of Chicago, Chicago, Illinois	June 23–July 18
Case Institute of Technology, Cleveland, Ohio	June 16–July 11
Remington Rand Building, Los Angeles, California	June 23–July 18
Remington Rand Building, New York, New York	June 9–July 3
University of Pennsylvania, Philadelphia, Pennsylvania	June 9–July 3
Remington Rand Building, Toronto, Canada	July 7–Aug. 1

Scientifically Oriented Courses

Southern Methodist University, Dallas, Texas	June 9–July 3
College of St. Thomas, St. Paul, Minnesota	Aug. 25–Sept. 5

Requests for information concerning this program should be directed to Remington Rand, Univac Educational Department, 315 Fourth Avenue, New York 10, New York.

• The University of Chicago School of Business has also announced a four week seminar beginning June 23 which will be conducted by the Univac Educational Dept. of Remington Rand. A new textbook and other material have been developed for the seminar, and the UNIVAC computer on campus will be available as an aid for teaching and discussion.

• The third annual Machine Records and Computing Conference for Educational Institutions was held at Oklahoma State University campus April 14–16. The conference committee included: Chairman of Registration Sessions, John Chaney of the University of Illinois; Chairman of Business Office Sessions, Kenneth Manning of the University of Michigan; Chairman of Computing Sessions, Walter C. Jacob of the University of Illinois; and General Chairman, W. M. Usher of the Oklahoma State University. The topics included the application of tabulating equipment to institutional registration procedures, permanent record posting, grade reporting and processing, and admission procedures. Other subjects included accounting and budgetary applications, administrative cost studies, the use of computers in research, statistics and engineering, organization of institutional computing centers, and reports on the university programs of equipment manufacturers.

• The Education Committee of the Los Angeles ACM Chapter has submitted a list of speakers to the office of the School and Industry Program, Los Angeles City Board of Education. These speakers were made available for programs to be held in connection with Boys' Week the latter part of April. It was planned to describe the development of computing machinery and the opportunities for future employment in the computing field.

COOPERATIVE PROGRAMMING GROUPS

GUIDE

The Agenda Steering Committee of GUIDE met in St. Louis April 3–4 to plan the forthcoming sessions at the Rice Hotel in Houston, May 7–9. The agenda will feature concurrent sessions in the programming area, the operations and equipment area, and the administrative area.

A GUIDE committee is meeting April 14–18 in New York City to attempt to delineate specifications for a business language amenable to an automatic coding system.

USE

The next meeting is scheduled for July 30 to August 1 in San Francisco.

ALWAC Users Association

The next meeting is scheduled for May 14–16 at the Reliance Electric Company in Cleveland, Ohio.

SHARE

A report of the Universal Language Committee has been distributed to all members. The committee work was motivated by the recent discussions in ACM of formulating a universal programming language. This committee also intends to provide suggestions to the committee representing the United States in the Geneva, Switzerland international meeting on universal programming language.

The report describes a language hierarchy in which three levels of language are specified. The first level is called POL (Problem Oriented Language). This is the most sophisticated language and would encompass most of the current codes now in use, such as Fortran, IT, Autocoder, etc. The second language level was called UNCOL (Universal Computer Oriented Language). This would be the universal language or switchbox. At present no such concept exists except in very primitive states. The third and last level was called ML (Machine Language). This language represents the conventional machine order code language but is universal in character.

The report also describes a technique involving translators between the various language levels. With this technique the translation for a given machine of a new problem oriented language to the machine language of the given machine can be accomplished automatically. Also the preparation of translation

programs to go from one language level to another, as would be necessary for an installation receiving a new computer, can be accomplished automatically with this technique. The development of this report is an outgrowth of meetings of the SHARE committee held in Washington, D. C. recently and the efforts of a number of Los Angeles 704 users.

NEWS ITEMS

• The Pegasus computer built by Ferranti, Ltd. of England has a magnetic drum of capacity 4,096 words and an additional 1,024 words storage of permanent storage for input-output routines and checking programs. The computer also has available 48 immediate-access single word registers and seven accumulators which are of the nickel delay line type. The word length of the computer is 39 binary digits. The computer involves single address logic and stores two instructions per word.

• Ferranti, Ltd. of England has announced a new computer called the "Mercury." It utilizes floating point arithmetic and has a computing storage of the magnetic core type with a capacity of 1,024 words. Backing this computing storage is a magnetic drum storage system in which 16,000 words can be recorded. Included in the design of the computer are seven B registers; instructions involving the B registers are carried out at considerably greater speed than arithmetic instructions due to a special arrangement which allows selection of only a part of a word out of magnetic core storage. Delivery of the first Mercury computers is imminent.

• A card reader which will read punched cards at a maximum speed of 2,000 cards per minute has been announced by the Uptime Corporation of Los Angeles. A model has been built and tested. The manufacturers claim excellent anti-jamming characteristics. Timing is done photoelectronically, and in no case is there any mechanical timing. The equipment is available for the 704 and the 1103A computers, and rental and purchase arrangements are possible.

• SAPO, a Czechoslovakian relay computer, employs a 5-address, two words per instruction logic. The first three addresses are used for the operands and the result location. The fourth and fifth addresses specify branch points for each instruction depending on the sign of the result. The computer has a drum memory and operates at about 3 instructions per second. (Courtesy Datamation)

• Stromberg-Carlson's Charactron-shaped beam electron tube and the Haloid Company's Xerox method of rapid printing to produce the world's fastest print-out of computer data ought to be a bottleneck-breaker in getting missile data faster. Numbers, letters and symbols on the Charactron's tube face are optically projected onto a Xerox printer drum. Result is a 4,680 line-per-minute printing capability. This is up to ten times faster than electro-mechanical printing (Courtesy San Diego Computer Newsletter)

• Daytronic Corporation of Dayton, Ohio offers a new unit, the Digital Indicator, designated the Model 600, which utilizes the new "synchro-sweep" principle in which a rotary potentiometer connected in a bridge circuit with the transducer, is driven continuously by a synchronous motor at a speed in synchronism with the reducer carrier frequency. The unit displays the transducer output on an electrical decade counter.

Adaption is easily made for employing the output with printers, electrical typewriters, punch card equipment, and digital computers. By the use of external scanning equipment, the outputs of several transducers can be displayed in rotation. (Courtesy San Diego Computer Newsletter)

• Packard Bell Computer Corporation has scheduled May 15 as the delivery date of TRICE (Transistorized Realtime Incremental Computer—Expandable) to the Computation Laboratories at Army Ballistic Missile Agency, Redstone Arsenal, Huntsville, Alabama. This new completely solid-state high speed computer can generate stable sine waves at rates as high as 8,000 cycles per second in real time. The machine is a parallel DDA type of computer; it iterates 100,000 times per second in parallel. It is modular in concept to the effect that a larger problem only requires additional integrators, multipliers, comparators, etc. Increasing the size of the problem, or machine, does not reduce the speed of computation since each module is its own arithmetic unit. Programming is done by means of a plugboard.

• An electronic outline of world history from 4 B.C. to the present, ranging from the birth of Christ

to the launching of Sputnik I, was demonstrated recently by International Business Machines Corporation in cooperation with the U. S. Department of State. The automated history text is stored in an IBM 305 RAMAC computer in ten languages—English, French, Italian, Nederlands, Spanish, Swedish, Portuguese, German, Russian, and Interlingua (a universal scientific language). Any annual portion of the history may be located by RAMAC in less than 2/3 of a second. Examples: "1480—Leonardo da Vinci invented the parachute. Dr. Johann Faustus, necromancer, was born in Germany. 1766—Mozart composed his first opera at the age of eleven. Christian VII became King of Denmark." The historical file was demonstrated to reporters by Dr. Cuthbert C. Hurd, IBM's Director of Automation Research.

The RAMAC "history book" will be a feature exhibit of the U. S. Pavilion at the Brussels World's Fair. A RAMAC computer, built in the new IBM plant at San Jose, California, especially for the Fair, was shipped to Brussels by cargo plane over the polar route in time for the opening on April 17. (Courtesy San Diego Computer Newsletter)

• General Kinetics Incorporated has recently completed a study of the use of commercially available digital computers in Air Traffic Control. The results show that reliability is the chief remaining obstacle to the construction of the semi-automatic ATC systems envisioned by the Airways Modernization Board. The new transistorized computers are expected to prove satisfactory, thus making a special machine unnecessary.

• An electric automatic splicer capable of splicing digital tapes, paper, magnetic films and tapes has been developed by Prestoseal Manufacturing Corporation of Long Island City, New York. The splicer utilizes controlled heat and pressure applied within an automatic time cycle; it does not require cement or adhesives. Paper tape uses the overlap method; tape (mylar or acetate) may be overlapped or butt-welded. Fusing is accomplished within 2¾ seconds. The unit is 9″ x 11″ x 17″ in size, weighs only 17 pounds, and operates on 110/120 volts a-c.

• The New York Office of the Service Bureau Corporation has added a data plotter to the 704 installation there, to be used for both scientific and business computing. It is believed that the use of such a plotter for commercial charts and graphs in this manner is a pioneering application in this field.

• Electronic Associates, Inc., plans to display its new 231-R analog computer at the Western Joint Computer Conference. The computer features 100 amplifiers and a number of other major improvements, including a new electronic digital voltmeter and an accessory Automatic Digital Input-Output System (ADIOS).

• Rocketdyne expects an IBM 709 in October, and North American's Los Angeles Division will follow suit in December.

• Litton Industries is adding two magnetic tape transports and a magnetic tape buffer to its ALWAC computer in April.

• Translated digests of the current Soviet electronic, automation, and physics literature have been announced for publication by International Physical Index, Inc., New York. Coverage of Soviet development in each special field will be achieved by the scanning of more than 60 Russian language professional journals. (Courtesy San Diego Computer Newsletter)

• Remington Rand Univac announces that the Air Materiel Command of the Air Force has placed firm orders for four UNIVAC Scientific Model 1105 computer systems. These computers will be installed at prime AMC depots at Rome, New York; Albany, Georgia; Sacramento, California and Dayton, Ohio. The first computer is scheduled for installation in August, 1958 at Albany, Georgia.

• A total of eight UNIVAC file computers are now installed at Douglas Aircraft Company in Santa Monica. Six of these computers are Model 0 types and two are Model 1 types.

• White Sands Proving Grounds of the U. S. Army has installed a UNIVAC Scientific 1103A computer as a replacement for the 1103 which it had for approximately two years.

• At Johns Hopkins University, the Applied Physics Laboratory's Computing Center headed by Dr. R. P. Rich is working on APT, a mathematical language compiler for the 1103A and 650 computers which combines the best features of IT with the efficiency of the Polish notation. It will be completed in a few months and should prove unusually fast and versatile. In a recent reorganization, Dr. Rich placed Charles Bitterli in charge of programming and Lowell McClung in charge of operations.

• The Johns Hopkins Operations Research Office Computing Center, headed by Mr. George Clark, has just completed the development of its ORO Service Library System for the 1103A computer which recently replaced its 1103. The work of the Center is mainly operational simulation and war gaming.

• Egon Weiss, co-author with William Ruleau of "The 650 Space Saver" while at ERCO, is now with the Applied Physics Laboratory at Johns Hopkins. William Ruleau is with Technical Operations, Incorporated.

• Harold Luxenberg has been named Technical Assistant to the Director of Engineering at Litton Industries in Beverly Hills, California.

• ALWAC Division of El-Tronics, Inc., has announced the following appointments: Victor White, Manager of the Computer Department and Robert Jackson, Chief Research Engineer. Both will report to Robert Cole, General Manager of the ALWAC Division.

• Dorothy Shumway, ACM member, who is with the Computer Lab at Convair, was recently nominated for the Woman-Of-The-Year Award given by the California Wing Auxiliary of the Air Force Association. Only one other woman from San Diego was nominated.

• Charles Falkner is leaving Convair, San Diego to join the Los Alamos Laboratory.

• In connection with the Los Angeles ACM's education activities, Ben Handy of Litton Industries spoke at David Starr Jordan High School assembly sponsored by the California Scholarship Federation and the high school's mathematics department; and Dr. Robert Rector of Space Technology Laboratories spoke at Los Angeles City College on "The Mathematical Elements of Computer Coding."

• William J. Pegg, Los Angeles ACM member, has recently established a Management Consulting firm offering services in commercial and scientific computer applications, management systems analysis, and classes in data processing.

• Robert Robinson, formerly of North American, is now at Litton Industries.

• Kenneth Wallingford, previously associated with Lockheed Aircraft, has joined ElectroData. Irvin Marshall, who has most recently been located at North American's Missiles Development Division, is now also on the staff of ElectroData in Pasadena.

• R. Berry and J. Kleinbard have moved to System Development Corporation from the Los Angeles Division of North American.

• New personnel at Space Technology Laboratories include Eugene Walsh from Detroit; A. Greenlee from San Diego; Jack Sacks from the Naval Ordnance Laboratory, Corona; David Giedt and Sherman Starr from North American's Los Angeles Division; Robert Finkel from Lockheed; and J. L. Sevy from the Missiles Development Division of North American.

• (Unofficial Report; Norden-Ketay is merging with United Aircraft.)

• L'Association Française de Calcul announces the publication of its new quarterly journal, "Chiffres." The first issue is dated March, 1958. Price per issue is 600 francs. The address of the Association is 98 bis, Boulevard Arago, Paris (XIV^{e}). Requests for subscription to the journal may be addressed to the Treasurer, M. P. Rapin, 5, Rue Général-Lanrezac, Neuilly (Seine).

• The Department of Commerce announced that it has recently requested a special appropriation of $300,000 to set up a Foreign Technical Information Center. In addition, the President's budget for 1959 included $1,250,000 for the Department's Foreign Technical Information Program. The new Center is intended to act as a central clearinghouse for the collection, the evaluation, and the distribution of foreign and scientific and technical literature for the use of American scientists and engineers. Plans have been worked out for the public distribution of information from such organizations as the National Science Foundation, the Atomic Energy Commission, the armed services and the intelligence agencies. Of particular interest to mathematicians is the fact that arrangements have been made to obtain, from these and other agencies, copies of abstracts and translations of foreign technical articles, monographs, and books which have been prepared by the agencies. It is estimated that these will pass through the Center at an annual rate of 50,000 abstracts and 10,000 complete translations.

The news release which announced the plans for the Center also made the following statement: "The new Center will operate a coordination service to eliminate duplication of translating among U. S. public and private agencies and by friendly foreign governments."

• The 1958 Western Joint Computer Conference exhibits were the largest in JCC history in amount

of money received and in size. Eighty-two booths were sold, and it was necessary to refuse some requests due to lack of booth space. Approximately fifty firms were represented, including research firms, manufacturers of digital and analog computers, component manufacturers and computer supply firms. All phases of computer technology were represented, including working computer systems, materials, and components, input-output equipment and storage systems. There was also a Sponsors' (AIEE, IRE, and ACM) Booth and an Exhibits Information Booth. The arrangements were handled by Exhibits Chairman David Weinberg of Space Technology Laboratories who was assisted by Edward Ward and Nancy Shaffer, also of STL.

• Several persons in the computer field in the Washington, D. C. area have expressed an interest in forming an informal group for the purpose of exchanging information and experience and at the same time providing a social meeting place for those interested in digital computing.

Using the Digital Computer Association of Greater Los Angeles as a model, a group is being formed to be known as DC^2 (District of Columbia Digital Computers). The tentative plan is a non-dues paying group which will meet monthly for cocktails, dinner and program. The program would consist of a speech, debate, and/or discussion on some phase of computing. The only semblance of organization would be a chairman, secretary-treasurer and program chairman. The aim would be to keep the association informal and independent but reasonably dignified and orderly. Persons interested may contact Mr. Howard Morrison at the Corporation for Economic and Industrial Research in Arlington, Virginia.

• The College of Engineering at New York University and the International Business Machines Corporation are jointly sponsoring a Symposium on Digital Computing in Chemical and Petrochemical Industries on May 13–14, 1958 at the Gould Student Center, University Heights Campus. This session is co-sponsored by the New York Section of the A.I.Ch.E and SIAM. Approximately sixteen papers will be presented. Contact: Professor M. A. Woodbury, New York University, New York 53, New York.

• The Proceedings of the National Simulation Council will soon be published for the first time. Approximately eighteen papers concerning analog simulation will be included. Copies may be ordered by sending a check for $3.00 to any of the eight Simulation Councils throughout the country. For further information, contact J. E. Sherman, Member of the National Simulation Council Board of Directors, at Lockheed Aircraft Corporation, Missile Systems Division, Research and Development Branch, Palo Alto, Calif.

• The proceedings of the Fourth Annual Computer Applications Symposium, held October 24–25, 1957, are now available from the Armour Research Foundation of Illinois Institute of Technology, 10 West 35th Street, Chicago 16, Illinois. The papers included in the proceedings cover computer applications in such areas as hospital insurance, air line reservation systems, transportation facilities, an air defense simulation, fluid flow, the 1960 census, and product development research. A recent announcement schedules the Fifth Annual Computer Applications Symposium for October 29–30, 1958 in Chicago, Illinois.

• The Northwest Computing Association will hold an International Computing Conference at the University of British Columbia, Vancouver, B. C. on May 24, 1958. The theme will be "The Exploitation of Small Computers in the Northwest." The speakers will include: H. Etkin of B. C. Electric, T. E. Hull of U. B. C., R. F. Julius of Western Computing Co., Ltd., Vancouver, J. B. Ward of Pacific Power and Light Co., Portland, R. J. Hansen of the Washington State Dept. of Highways, C. Blake and R. J. Brown of the Bonneville Power Administration, C. Taft of IBM, D. E. Kibler of Shell Oil Co., and H. E. Tellier of General Electric Hanford Atomic Products Operation. Subject matter will vary all the way from "The Teaching Aspects of Computers and Numerical Analysis" to "The Blending of Poultry and Pigs Feed." The ALWAC III-E of the University of British Columbia Computing Centre is adjacent to the vendor display area and will be on display during the conference.

• The 13th National Meeting of the ACM at the University of Illinois, Urbana, Illinois, June 11–13, will be conducted in two parallel sessions, with fifteen-minute contributed papers and thirty-five-minute invited papers in both sessions. Tentative plans for the program include: a panel discussion on matrix computation problems (questions may be submitted in advance) and papers on round-off considerations in fixed and floating point arithmetic, the role of universities in the design and construction of computers, satellite orbit calculations, non-linear programming, partial differential equations, business

applications, and human and computer languages. Preprints of papers will be available at the time of registration, June 11. There will be no equipment exhibits. A cocktail party and banquet are planned.

• A symposium and exhibits on Documentation for a Study of Information Retrieval Systems was sponsored by the USC School of Library Science April 9–11. The symposium featured such widely recognized speakers as H. R. J. Grosch, Harley Tillitt, H. P. Luhn, Peter Worsley, Finley Carter, and Willis Ware.

• The Washington, D. C. section of SIAM is sponsoring a discussion group on automatic programming under the chairmanship of Tom Cheatham of Technical Operations. The first meeting of this group, a dinner meeting on March 25, was attended by 22 people. A review of interpretive programs for the Bendix G-15 computer was given by Robert Sweet, Bendix Eastern Regional Manager. The program announced for the April 30 dinner meeting concerned the specifications and design of compilers being developed by three local research organizations. Two of these compilers are for the 704 and the third for both the 650 and 1103A. The May meeting will deal with the problem of a common language.

• On March 13 a Simulation Council meeting was held at Point Mugu, California on the subject "The Use of a Flight Table in a Simulation Facility." The speakers were W. Uplinger (NOTS, Point Mugu), A. Mauldin (NOTS, Pasadena), R. W. Bond (Autonetics Div. of North American), P. Kelly (Convair Astronautics, San Diego), and D. Lipscomb (Nortronics Div. of Northrop). A tour of the installation was arranged by Willard Uplinger of the host facility.

• At the April meeting of the Digital Computers Association in Los Angeles, George Vosatka, Western Regional Manager of Bendix Computer, discussed "A Place for the Satellite Computer." He analyzed the need for and use of small-to-medium scale computers as supporting systems for large-scale installations, the desirable characteristics of a satellite computer, and the use of the Bendix G-15 as a supporting system.

• A special demonstration by Bendix Computer Division in Los Angeles indicated how computer equipment now available can handle income tax returns on a mass production basis. A Bendix G-15 was programmed to calculate the return, decide whether standard or itemized deductions should be taken, and type the results on a standard 1040 form. At the demonstration, Los Angeles newspapermen were invited to let the computer handle their 1957 tax computation and processing.

• Electronic Associates, Inc., gave a two-week computer applications course beginning April 28. They were recently host to a meeting of the American Institute of Chemical Engineers.

• IBM held a Western Region Applied Science Representatives meeting at the UCLA Residential Conference Center at Lake Arrowhead, California, April 13–16. Attendees came from as far away as Seattle, Washington, and Shreveport, Louisiana. George Brown of the Western Data Processing Center addressed the group.

• At a recent open house at Northrop Aircraft's new Science Center in Los Angeles, its IBM 704 was a major point of interest to many Northrop personnel. Specially built scale models, cartoon displays of the processing flow, and taped explanatory talks provide an impressive permanent display when viewed through the observation window of the computer room.

• The technical speakers at the Central Ohio Association for Computing Machinery "Symposium on Recent Advances in Programming Methods," on March 29 included: John Carr III of the University of Michigan, Frank Engel of Westinghouse Electric, J. H. Wegstein of National Bureau of Standards, H. R. J. Grosch of IBM, John Mauchley of Sperry Rand, and Everett Yowell of National Cash Register.

• The Finance Division of the American Management Association has announced a seminar at the Hotel Astor in New York City on May 14–16, "Planning Feasibility for EDP/IDP (Electronic and Integrated Data Processing). Case studies of business applications of an IBM-650, a Datatron, and a UNIVAC I will be included.

• The Case Institute of Technology has announced a short course in "Operations Research in Production and Inventory Control" June 2–13. The course will include such items as inventory theory, allocation theory, sequencing theory, routing theory, queuing theory, replacement theory, linear and non-linear programming, dynamic programming, quadratic programming, and other extremal methods. Case histories of business and industry applications will be used extensively. Contact: Dr. E. L. Arnoff, Operations Research Group, Case Institute of Technology, 10900 Euclid Avenue, Cleveland 6, Ohio.

COMING EVENTS

- "A Company President Looks at Electronic Data Processing"
  May 1, 2, 1958; University of California Residence Conference Center, Lake Arrowhead, California
  Sponsor: Western Data Processing Center and UCLA
  Contact: Sam Houston, UCLA Engineering Extension, Rm 3116, West Los Angeles, California
- Fourth National Flight Test Symposium
  May 4–7, 1958; Park Sheraton Hotel, New York City, New York
  Sponsor: ISA
- National Symposium on Microwave Theory and Techniques
  May 5–7, 1958; Stanford University, Stanford, California
- Western Joint Computer Conference—"Contrasts in Computers"
  May 6–8, 1958; Ambassador Hotel, Los Angeles, California
  Contact: Dr. Willis Ware, The RAND Corp., Santa Monica, California
- GUIDE Meeting
  May 7–9, 1958; Rice Hotel, Houston, Texas
- Los Angeles Chapter ACM Symposium "Small Automatic Computers and Input-Output Equipment—A Report from the Manufacturers"
  May 9, 1958 (the day following WJCC); Ambassador Hotel, Los Angeles, California
  Contact: P. Armer, The RAND Corporation, Santa Monica, California
- UNIVAC Users Conference
  May 12–13, 1958; Pacific Mutual Life Insurance Company, Los Angeles, California
- Symposium on Digital Computing in Chemical and Petrochemical Industries
  May 13–14, 1958; Gould Student Center, University Heights Campus of New York University, New York
  Sponsor: NYU College of Engineering, IBM, A. I. Ch. E. (New York Section), and SIAM (New York Section)
  Contact: Professor M. A. Woodbury, New York University, N. Y. 53, N. Y.
- ALWAC Users' Association Meeting
  May 14–16, 1958; Reliance Electric Company, Cleveland, Ohio
- Northwest Computing Association International Computing Conference
  May 24, 1958; University of British Columbia, Vancouver, B. C.
- 1958 National Telemetering Conference
  June 2–4, 1958; Lord Baltimore Hotel, Baltimore, Maryland
  Sponsors: ISA, IAS, AIEE
- Computer and Automation Conference
  June 2–4, 1958; University of Texas, Austin, Texas
  Sponsor: University of Texas
- Canadian Conference for Computing and Data Processing
  June 9–10, 1958; University of Toronto, Toronto, Ontario, Canada
  Contact: Mr. H. W. Rowlands, 15 Wellington St. W., Toronto, Ontario
- Fourth International Automation Exposition and Congress
  June 9–13, 1958; Coliseum, New York City, New York
- 1958 ACM National Conference
  June 11, 12, 13, 1958; University of Illinois, Urbana, Illinois
- American Mathematical Society Meeting
  June 20, 1958; Corvallis, Oregon
- Fifth Annual Symposium on Computers and Data Processing
  July 24–25, 1958; Denver Research Institute, Denver, Colorado
  Sponsor: Denver Research Institute
  Contact: C. A. Hedberg, Denver Research Institute

- USE Meeting
  July 30–August 1, 1958; San Francisco, California
- International Congress of Mathematicians
  August 14–21, 1958; Edinburgh, Scotland
- WESCON
  August 19–22, 1958; Ambassador Hotel and Pan Pacific Auditorium, Los Angeles, California
- American Mathematical Society—63rd Summer Meeting
  August 25–30, 1958; Cambridge, Massachusetts
- The Mathematical Association of America—39th Summer Meeting
  August 25–28, 1958; Cambridge, Massachusetts
- SHARE Meeting
  September 10–12, 1958; San Francisco, California
- 1958 National Simulation Conference
  October 23–25, 1958; Statler-Hilton Hotel, Dallas, Texas
  Sponsors: IRE-PGEC and Dallas Section of IRE
  Contact: J. E. Howard, 2100 Menefee Drive, Arlington, Texas
- American Mathematical Society Meeting
  October 25, 1958; Princeton University, Princeton, New Jersey
- Fifth Annual Computer Applications Symposium
  October 29–30, 1958; Chicago, Illinois
  Sponsor: Armour Research Foundation, Chicago, Illinois
- International Conference on Scientific Information
  November 16–21, 1958; Mayflower Hotel, Washington, D. C.
  Sponsor: National Academy of Sciences, National Research Council, National Science Foundation, American Documentation Institute
  Contact: Secretariat, International Conference on Scientific Information, National Academy of Sciences, 2101 Constitution Avenue, N. W., Washington 25, D. C.
- American Mathematical Society Meetings
  November 20–21, 1958; Pomona, California
  and
  November 28–29, 1958; Northwestern University, Evanston, Illinois
  and
  November 28–29, 1958; Durham, North Carolina
- Eastern Joint Computer Conference
  December, 1958; Philadelphia, Pennsylvania
- American Mathematical Society—65th Annual Meeting
  January 20–22, 1959; University of Pennsylvania, Philadelphia, Pennsylvania
- Joint Meeting of Institute of Mathematical Statistics (Central Region) and the Association for Computing Machinery
  April 2–4, 1959; Case Institute of Technology, Cleveland, Ohio
  Contact for IMS: Martin B. Wilk, Bell Telephone Laboratories, Murray Hill, New Jersey
  Contact for ACM: Daniel Teichroew, National Cash Register, Dayton 9, Ohio
- 1959 ACM National Conference
  Summer, 1959; Massachusetts Institute of Technology, Cambridge, Massachusetts
  Contact: F. Verzuh, MIT
- American Mathematical Society Meeting
  Summer, 1959; Salt Lake City, Utah

A NEW CHALLENGE IN DATA PROCESSING...

The Astro-Electronic Products Division of the Radio Corporation of America, Princeton, N. J., has immediate opportunities in an advanced and challenging area of automatic data processing. The long-range research program now beginning will involve the solution of logic problems and the programming of deductive rules (non-numeric).

Technical proficiencies are needed in information-systems analysis, data processing programming research, machine logical designs, applied probability theory, machine language-translation, and automatic filing systems.

To participate in this program applicants must have advanced technical degrees, or equivalent experience, and a record of achievement. A government security clearance is required of all project personnel.

These opportunities offer stimulating positions in an atmosphere conducive to creative activity. For a personal interview with Mr. M. S. Cohen, Project Director, call collect or send resume to:

Mr. D. D. Brodhead
Astro-Electronic Products Division
RADIO CORPORATION OF AMERICA
Princeton, N. J.
WAlnut 4-2700, Ext. 706

Members of ASSOCIATION for COMPUTING MACHINERY:
For Personal Interview at Urbana, Illinois Meeting, June 11, 12, 13, look for RCA Representative, Mr. D. D. Brodhead.

EQUAL OPPORTUNITY For All Qualified Applicants

RCA®

RADIO CORPORATION of AMERICA
ASTRO-ELECTRONIC PRODUCTS DIVISION
Princeton, New Jersey